إصنع بنفسك ..

Do It Your Self



Microsoft Visual Basic 2013



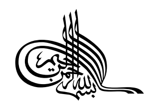
المركز الرئيسي : 11 شارع د/ محمد رأفت – محطة الرمل – الإسكندرية
تليفون وفاكس: 4838326 (03)(2+)
موبايل : 01001634294 (2+)

www.daralbraa.com
info@darAlbraa.comEmail:

محمد خميس عبد الرازق
دراسات عليا حاسب آلى ونظم المعلومات
mkh_eg60@yahoo.com



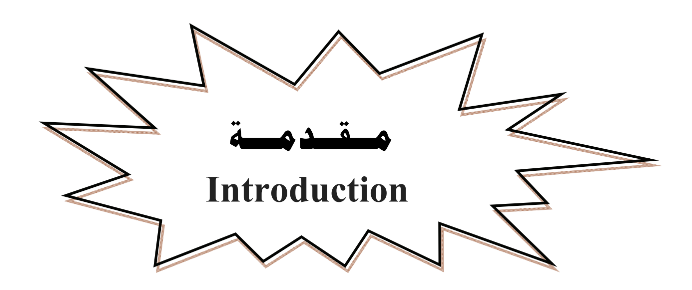
الناشر :	دار البراء لنشر وتوزيع الكتب العلمية
رئيس مجلس الإدارة :	إبراهيم محمد إبراهيم زبير
اسم الكتاب :	Visual Basic.Net 2013
المؤلف :	محمد خميس عبد الرازق
رقم الإيداع :	2846/2014
الترقيم الدولي :	978-977-723-018-6
المقاس :	24 x 17
عدد الصفحات :	288
العنوان :	11 شارع د/محمد رأفت – محطة الرمل – الإسكندرية
تليفون وفاكس :	(+2)(03) 4838326
للاستعلام والمبيعات :	(+2) 01001634294
البريد الالكتروني :	info@daralbraa.com
الموقع :	www.daralbraa.com



سُبْحَانَكَ لَا عِلْمَ لَنَا إِلَّا مَا عَلَّمْتَنَا إِنَّكَ أَنْتَ الْعَلِيمُ الْحَكِيمُ



طبعة 2014



البرمجة لها أهمية كبيرة فى مجال نظم وتكنولوجيا المعلومات .. فبدون البرمجة لن نستطيع تصميم وتشغيل البرامج المختلفة والتعامل معها ، وكذلك إمتد الأمر ليشمل مواقع الإنترنت المختلفة وبرمجيتها المتعددة .

فصناعة البرمجيات فى الوطن العربى هو أحد الإتجاهات الحديثة فى مجال نظم وتكنولوجيا المعلومات فى الوقت الحالى .. خاصة فى عصر ما وراء المعرفة ونحن فى منتصف العقد الثانى من الآلفية الثالثة .

لذلك نقدم لكم فى هذا الكتاب أساسيات البرمجة بلغة الـ Visual Basic . Net فى نقلة جديدة فى عالم البرمجيات .. نظراً لبساطة وسهولة ومرونة لغة الـ Visual Basic .. للمبتدئين فى مجال البرمجة .. حيث يتناول هذا الكتاب المبادئ الأساسية للغة الـ Visual Basic بالأمثلة والتطبيقات ، بالإضافة إلى تصميم وإدارة قواعد البيانات ، ومجموعة متنوعة من التطبيقات المختلفة .. لكى يستوعب القارئ بصفة عامة ، والدارس لهذا التخصص بصفة خاصة المهام المختلفة للبرمجة .

حيث وفرت فى هذا الكتاب مجموعة من الأفكار والأساليب البرمجية لتناسب كافة المستويات .. فهى بداية خطوة بخطوة للمبتدئين ، وإضافة معلومات متواضعة للمحترفين .. بالإضافة إلى تكوين صورة ذهنية لدى مستخدمى البرامج الجاهزة ، وبرامج قواعد البيانات عن .. كيف تم تصميم هذا البرنامج الذى نعمل عليه ؟

محمد خميس عبدالرازق
دراسات عليا حاسب آلى
ونظم المعلومات
(محاضر ومصمم نظم المعلومات الآلية)



قبل الدخول فى عالم البرمجة بلغة الـ Visual Basic .. نتعرف فى هذا الجزء على مجموعة من المهام والتعاريف ، وخطة العمل لرحلتنا فى هذا الكتاب لكى يستطيع القارئ إخراج التطبيقات بشكل إحترافى .. فهناك مجموعة من الأسئلة لابد من الإجابة عليها .. وهى :

■ لمن هذا الكتاب ؟

هذا الكتاب موجه للمهتمين بمجال البرمجة ، أو الراغبين فى التعرف على كيفية تصميم البرامج بصفة عامة .. والدارسين لتخصص أقسام الحاسبات ونظم المعلومات بالكليات والمعاهد المختلفة ليساعدهم فى إيضاح المادة العلمية بشكل مبسط ، ومساعدتهم أيضاً فى إخراج مشروعات التخرج خطوة بخطوة .

■ هل دراسة البرمجة أمر معقد ؟

بالطبع **لا** .. " **فطريق الآلف ميل يبدأ بخطوة واحدة** " .. فلا يوجد محترف بدون ممارسة هوايته حتى الإحتراف .. فالأمر بسيط كل ما عليك أن تقوم بفهم الموضوع والفكرة جيداً .. ثم تطبيق الأمثلة ، والتعلم من أخطائك التى تقع فيها ، فلا يوجد مبرمج محترف بدون أخطاء سابقة .

■ كيف يمكنك تصميم وبرمجة أى برنامج بلغة الـ Visual Basic ؟

إتباع الخطوات الموجودة بكل تطبيق خطوة بخطوة .. بالإضافة إلى إستخدام الأكواد البرمجية الموجودة على أسطوانة التطبيقات المرفقة مع هذا الكتاب .

■ ماهى لغة الـ Visual Basic ؟

هى لغة برمجة بسيطة تناسب المبتدئين فى مجال البرمجة ، ويستخدمها أيضاً المبرمجين المحترفين فى تصميم وبرمجة وإخراج البرامج بشكل إحترافى بسيط .

■ ماهى البرامج التى يمكن تصميمها بلغة الـ Visual Basic ؟

برنامج Windows Media Player ، وبرامج متصفح الإنترنت ، وبرنامج الآله الحاسبة Calculator ، ومجموعة برامج Office ، وبرامج قواعد البيانات Data Base ، وتصميم الواجهات الجذابة للبرامج Interface ، والعديد من البرامج التطبيقية الأخرى Applications .

■ كيف يمكنك التصميم والبرمجة بلغة الـ Visual Basic ؟

يمكن تصميم وبرمجة أى نظام ، أو أى برنامج من خلال إتباع مجموعة من الخطوات .. مثل :

1 – التفكير فى هدف البرنامج الذى تقوم بتصميه ، وما هى المهام التى يؤديها هذا البرنامج ؟

2 – تحليل النظام الخاص ببناء البرنامج ، وتحديد عدد النوافذ والوجهات التى يتكون منها البرنامج ، وتحديد طريقة عرض البرنامج على الشاشة .

3 – التفكير فى إمكانيات الشخص الذى سوف يستخدم البرنامج .. هل هو محترف ، أم مبتدئ ؟ .. ثم تصميم برنامج يناسب إحتياجاته .

4 – التعرف على العديد من البرامج الأخرى التى تم تصميمها ، والتعلم منها .

5 – إستخدم المرونة والبساطة فى التصميم .. ليكون البرنامج أكثر جاذبية .

■ هل لابد من حفظ الأكواد البرمجية الخاصة بلغة الـ Visual Basic ؟

بالطبع **لا** .. لا يتم حفظ الكود البرمجى نصاً .. ولكن فهم طريقة تركيب الكود داخل لغة البرمجة ، وطريقة كتابته داخل نافذة محرر الكود .. حيث يمكنك نسخ الأكواد المرفقة مع هذا الكتاب بدلاً من إعادة كتابتها .

■ هل الأكواد المرفقة مع هذا الكتاب كافية لإخراج العمل ؟

نعم .. فهى تلك الأكواد التى تم تصميم أمثلة وتطبيقات هذا الكتاب ، وأيضاً تصميم البرامج الملحقة مع هذا الكتاب .

■ لماذا يمتنع العديد من المهتمين بمجال نظم وتكنولوچيا المعلومات من الإقتراب من دراسة البرمجة ؟

السبب فى ذلك .. هو حالة الجهل ، والفشل الثقافى الموجودة فى سوق تكنولوچيا المعلومات خاصة فى الدول المستهلكة للتكنولوچيا .. والتى تتمثل فى :

- بعض الكتب والمراجع الخاصة بالبرمجة مجهولة المصدر ، والتى تحتوى على محتوى علمى دون المستوى المناسب لعقل الدارس لهذا المجال .

- بعض مواقع الإنترنت التى تحتوى على معلومات عن البرمجة وبها أخطاء علمية فادحة دون إشراف من قبل المتخصصين فى مجال الحاسبات والمعلومات .

- إدعاء بعض الأفراد ذوى الخبرة القليلة بالمعرفة ، وحالة الإرتجال العشوائى خاصة تكرار الكلام المستهلك والمتداول بأخطائه على الإنترنت .

- تضخيم بعض الأفراد لمحتوى المادة العلمية غير الهادف فى النهاية .

وبعد أن تعرفنا على أهم الأسئلة التى تدور فى ذهن أى فرد تجاه هذا الموضوع .. نتعرف الأن على سيناريو العمل داخل هذا الكتاب ، ومحتوى الفصول المقررة ..

■ الفصل الأول / مقدمة فى البرمجة

نتعلم فى هذا الفصل :

- ما هو جهاز الكمبيوتر ؟ وما هى وحدات ومكونات جهاز الكمبيوتر ؟
- ما هى أنظمة التشغيل المحتلفة التى تعمل عليها البرامج الجاهزة ؟
- بداية فكرة عمل البرمجة .. فهى تشبه محاكاة الإنسان للواقع المحيط .
- فكرة عمل جهاز الكمبيوتر .. والتى تتمثل فى (إدخال البيانات / التخزين فى الذاكرة / إمكانية تنفيذ الأوامر / إخراج النتائج) .
- فكرة عمل البرمجة المسيرة بالأحداث EDP .. فهى مجموعة من الأحداث المتتابعة ، والتى تكون فى النهاية شكل البرنامج الذى نتعامل معه .
- فكرة البرمجة بالكائنات OOP .
- مراحل إنشاء أى برنامج .. وهى (التحليل / التصميم / البرمجة / إختبار البرنامج / التنفيذ والتطبيق) .

■ الفصل الثانى / التعرف على بيئة الـ Visual Basic

نتعلم فى هذا الفصل :

- خطوات تحميل برنامج Microsoft Visual Studio .. من موقع شركة Microsoft .
- خطوات إعداد البرنامج Setup على جهاز الكمبيوتر .
- خطوات تشغيل البرنامج ، والتعرف على واجهة إستخدام البرنامج .. والتى تتمثل فى :
  - النماذج Forms .
  - صندوق (شريط) الأدوات Tool Box .
  - نافذة الخصائص Properties .
  - الأحداث Events .
  - الإجراءات Procedure .
  - الأساليب Methods .

■ الفصل الثالث / كتابة الكود داخل لغة الـ Visual Basic

نتعلم فى هذا الفصل :

- ما هو الكود البرمجى ؟
- التعرف على نافذة كتابة الكود Code Editor .
- القوائم Menu .
- العمليات الحسابية داخل لغة الـ Visual Basic .

■ الفصل الرابع / التعامل مع المشروع Project

نتعلم فى هذا الفصل :

- المشروع .. هو البرنامج فى مرحلة التصميم .
- كيفية إنشاء مشروع جديد New Project .
- إضافة نموذج جديد للمشروع الحالى .
- إضافة ملف مخزن سابقاً للمشروع الحالى .
- فتح مشروع موجود لمتابعة وإستكمال العمل عليه ، أو تعديله .
- حفظ ملفات المشروع الداخلية .
- كيفية تصميم واجهة التطبيق للبرامج .
- خصائص النماذج Forms ، والأدوات Tools ، والأحداث Events .

■ الفصل الخامس / التعامل مع صندوق الأدوات Tool Box

نتعلم فى هذا الفصل :

- صندوق (شريط) الأدوات .. وهو الذى يوجد بداخله كل الأدوات التى يتكون منها البرنامج فى النهاية .. مثل (المفاتيح / حقول الإدخال / مربعات وقوائم الإختيارات) .

- أمثلة على كيفية تكوين بداية البرامج .. خطوة بخطوة .

■ الفصل السادس / المتغيرات والثوابت

نتعلم فى هذا الفصل :

- المتغير .. هو مكان يتم حجزه فى الذاكرة ، ويتم الإعلان عنه بواسطة المبرمج داخل الكود .. وتتغير قيمته أثناء التشغيل .. مثل تعريف أسم معين غير معروف فى البرنامج .

- التعرف على أنواع المتغيرات (البيانات) .
- الثوابت .. هى بيانات ثابتة لا تتغير قيمتها أثناء تشغيل البرنامج .
- التعليقات Comments .. هى ملاحظات توضح شرح الكود المكتوب لعرض وظيفة معينة .

■ الفصل السابع / المصفوفات Arrays

نتعلم فى هذا الفصل :

- المصفوفة .. هى وسيلة لتخزين مجموعة من القيم ، أو البيانات التى لها نفس النوع .. وهى تشبه بيانات أعمدة الجداول فى قواعد البيانات .. مثل (الأسم / العنوان) .
- التعرف على المصفوفات المتغيرة ، والمصفوفات متعددة الأبعاد .

■ الفصل الثامن / الجمل الشرطية والحلقات التكرارية

نتعلم فى هذا الفصل :

- الجمل الشرطية .. تستخدم لإختبار شرط معين فى البرنامج سواء تحقق أم لا من خلال مجموعة من الجمل الشرطية .. مثل :
(Select Case / IF .. Then .. Else / IF .. Then)

• الحلقات التكرارية Loops .. هى حلقات من الجمل تسمح بتنفيذ شرط معين عدة مرات .. مثل الحلقات :
(While End While / Do Loops / For .. Next)

■ الفصل التاسع / التعامل مع الدوال Functions

نتعلم فى هذا الفصل :

• الدالة .. هى التنبؤ بقيمة متغيرات مجهوله بدلالة متغيرات أخرى معلومة ، وذلك من خـلال مجموعة من الـدوال الجاهزة داخل لغة الـ Visual Basic .. مثل الدوال التالية :

- الدالة UCase .. لتحويل الحروف الصغيرة Small .
- الدالة LCase .. لتحويل الحروف الكبيرة Capital .
- الدالة StrConv .. تحويل الحروف .
- الدالة Len .. لمعرفة طول الحروف داخل النص .
- الدالة Replace .. لإستبدال مقطع من النص .
- الدالة StrReverse .. عكس حروف النص .
- الدالة Instr .. البحث عن قيم داخل النص .
- الدالة Trim .. حذف المسافات الموجودة يمين ، ويسار النص .
- الدالة Beep .. لإصدار صوت تنبيهى .
- الدالة Now .. لإيجاد الوقت والتاريخ الحالى .
- الدالة ToDay .. لإيجاد تاريخ اليوم فقط .

■ الفصل العاشر / إكتشاف وتصحيح الأخطاء

نتعلم فى هذا الفصل :

• كيفية إكتشاف الأخطاء Bug .
• تصحيح الإخطاء Debug .
• أنواع الأخطاء (التركيبية / التنفيذية / المنطقية) .

■ الفصل الحادى عشر / قواعد البيانات Data Base

نتعلم فى هذا الفصل :

• التعريف بقواعد البيانات ، ومكونات قاعدة البيانات .
• برامج قواعد البيانات .

- إنشاء قاعدة البيانات بإستخدام برنامج Microsoft Access .. (إنشاء الجداول " الملفات " / العلاقات بين الجداول / الإستعلامات / إدخال البيانات فى الجداول) .

- تصميم واجهات التطبيق ، وربطها بقاعدة البيانات .
- الإتصال بحقول قاعدة البيانات .
- كتابة الأكواد البرمجية .
- إخراج البرنامج للعمل من خلال أيقونة Setup .
- تشغيل وإدارة قاعدة البيانات .

■ **الفصل الثانى عشر / تطبيقات بإستخدام لغة الـ Visual Basic**

نتعلم فى هذا الفصل :

- تصميم متصفح إنترنت Web Browser .
- تصميم أله حاسبه Calculator .
- تصميم نافذة إفتتاحية لواجهة البرنامج Splash Screen .
- تصميم نافذة لإظهار كلمة سر البرنامج password Login Form .
- تصميم شريط القوائم Menu Bar داخل البرنامج .
- تصميم نص متحرك داخل نافذة البرنامج .

الفصل الأول

مقدمة فى البرمجة

Introduction to Programming

نتعلم فى هذا الفصل ..

- فكرة عمل جهاز الكمبيوتر .
- وحدات الـ Hardware ، والـ Software .
- التعريف بلغات البرمجة .
- بداية فكرة عمل البرمجة .
- المفسر ، والمترجم .
- البرمجة المسيرة بالأحداث EDP .
- البرمجة بالكائنات OOP .
- مراحل إنشاء أى برنامج بلغة الـ Visual Basic .

الفصل الأول

مقدمة فى البرمجة

Introduction to Programming

■ مقدمة :

فى البداية كان جهاز الكمبيوتر عبارة عن آله حسابية تقوم بعمليات (الجمع ، والطرح ، والضرب ، والقسمة) .. لكى يخرج الناتج فى صورته النهائية ، وكانت تستهلك هذه الآله كمية كبيرة من الطاقة الكهربائية .. بالإضافة إلى خروج كمية كبيرة من الطاقة الحرارية .

وبالتالى تطور جهاز الحاسب مع تطور الزمن ، وتطور الأشياء المحيطة حولنا .. وتعددت الأغراض المستخدمة من خلاله .. ليصبح جهاز الحاسب مكون من جزئين وهما :

• أولاً : الـ Hardware :

وهى الأجزاء (الوحدات) الملموسة المكونة لجهاز الحاسب مثل :

- الشاشة Monitor / Screen .
- لوحة المفاتيح Key Board .
- الفأرة Mouse .

• ثانياً : الـ Software :

وهى الأجزاء غير الملموسة المشغلة لجهاز الحاسب .. وتنقسم وحدات الـ Software إلى 3 أقسام وهى :

1 – نظام التشغيل Operating System :

وهو أداة الربط بين المستخدم User ، وبين جهاز الكمبيوتر .. مثل أنظمة :

- DOS .
- Windows .
- Linux .
- Unix .

2 – برامج التطبيقات Application Programs :

وهى برامج مصنعة خصيصاً لأداء وظيفة محددة لخدمة المستخدمين فى مجالات معينة .. مثل مجموعة البرامج المكتبية Office ، ومجموعة برامج الـ Graphics .. وغيرها من المجموعات البرمجية الجاهزة .

3 – لغات البرمجة Programming Language :

وهى لغة بناء البرامج الجاهزة وبرامج التطبيقات .. لكى تكون حوار جاهز بين المستخدم وجهاز الكمبيوتر .

- **إذن :**
  يعتبر الـ Hardware ، والـ Software وجهين لعملة واحدة .

■ بداية فكرة عمل البرمجة :

وبدأت فكرة البرمجة فى رغبة الإنسان فى التحكم فى جهاز الكمبيوتر لأداء وظائف محددة بخلاف العمليات الحسابية ، والعمليات المنطقية .. فبدأت نشأة فكرة الأوامر التى يستطيع المستخدم بها أن يطلب من الكمبيوتر أداء وظائف معينة فيؤديها .

ولأن جهاز الكمبيوتر لا يفهم أى شئ إلا الشحنات الكهربائية الموجودة بداخلة ونرمز لها بالرمز (1) .. ولو الشحنات الكهربائية غير موجودة نرمز لها بالرمز (0) .. ومن خلال مجموعة الصفر والواحد (000111) إستطاع الإنسان كتابة أول برنامج للحاسب الآلى .. (أى مجموعة من الأوامر المحددة) التى يرغب فيها المستخدم .

- وتسمى هذه العملية **بالبرمجة Programming** .. وتسمى هذه اللغة بلغة الألة Machine Language .. وهى لغة تعتمد على النظام العددى الثنائى Binary System والذى يتكون من رقمين فقط هما (0 ، 1) .

- ومن خلال عمليات التطوير المستمرة فى الحياة إستطاع الإنسان تطوير لغات البرمجة من خلال إدخال بعض المفردات الدارجة .. وبالتالى تزداد صعوبة كتابة وفهم البرامج المكتوبة باللغات ذات المستوى الأدنى . وهى (لغة الأله)

- إستمرت عملية التطوير حتى أصبح الإنسان يستطيع أن يكتب برنامج كامل بإستخدام مفردات لغتة .. لتتطور لغات البرمجة لتصبح أقرب ما يكون من لغة الإنسان (محاكاة العنصر البشرى) .

- إذن .. لغات البرمجة = لغات الإنسان .

وتسمى هذه اللغات (لغات المستوى الأعلى High Level Language) ومن أمثلتها لغات :

- البيسك Basic .
- الكوبول Cobol .
- الباسكال Pascal .
- الفورتران Fortran .

- أما عن البرمجة بلغة الـ Basic .. بدأت فى إحدى المعامل فى الولايات المتحدة الأمريكية لتكون من أول وأسهل اللغات التى إستخدمها الإنسان .. لذلك فهى لغة تناسب المبتدئين فى مجال البرمجة .

- أما الآن .. فقد تطورت لغة الـ Basic كثيراً وتعددت إمكانيتها وزادت قدرتها ، وأصبحت من أشهر وأسهل وأقوى لغات البرمجة فى العالم .. وكان لابد من هذا التطور ليناسب رغبات المستخدم ، والمبرمج ، ونظم التشغيل ، وأسلوب البرمجة .

- **فمثلاً** .. الإصدار القديم من لغة الـ Basic كان يعمل تحت بيئة نظام التشغيل DOS .. أما الإصدارات الأخيره من الـ Basic وهو ما يسمى بالـ Visual Basic أى لغة البيسك المرئية تعمل تحت بيئة نظام تشغيل الـ Windows.. وهو محور عمل وموضوع هذا الكتاب .

س / ما هى فكرة عمل جهاز الكمبيوتر قبل عملية البرمجة ؟

تقوم الفكرة على مجموعة من الخطوات المتتالية كما يلى :

1 – إدخال (قراءة) الحاسب للبيانات Input :

يتم إدخال البيانات بواسطة المستخدم لجهاز الكمبيوتر ، وتحويلها إلى معلومات فيما بعد .

2 – تخزين البيانات فى الذاكرة الداخلية (المؤقتة RAM) :

حيث يقوم المبرمج بإعطاء أسماء محددة للأماكن التى يتم تخزين البيانات بها داخل ذاكرة الحاسب .. وعن طريق هذه الأسماء يستطيع البرنامج التعامل مع البيانات المخزنة فى هذه الأماكن .

- وتسمى هذه الأماكن بالمتغيرات Variable .. وهى عبارة عن قيم مخزنة تتغير أثناء التعامل مع البرنامج .

3 – إجراء العمليات الحسابية المختلفة على البيانات Arithmetic :

حيث يقوم جهاز الحاسب من خلال المعالج Processor بإجراء العمليات الحسابية (الجمع / الطرح / الضرب / القسمة) .. وذلك من خلال الأوامر التى يقوم المستخدم بتوجيهها لجهاز الحاسب .. كما يمكن لجهاز الحاسب أيضاً القيام بعمليات المقارنة بين المتغيرات .

4 – إجراء العمليات المنطقية Logical :

مثل الإختيار بين أحد بديلين ، أو أمرين .. ثم إختيار البديل الأفضل وذلك بعد إجراء عملية المقارنة لغرض إتخاذ القرار Decision .. مثل تقديرات نتائج الطلاب (ممتاز / جيد جداً / جيد / مقبول / ضعيف / ضعيف جداً) .

5 – إمكانية تكرار تنفيذ الأوامر Loop :

من المميزات الهامة لجهاز الكمبيوتر :

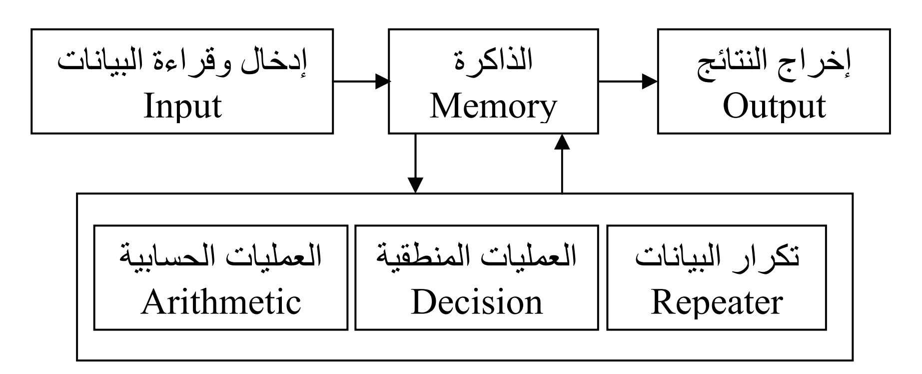
- أنه لا يشعر بالملل عند تنفيذ أى أمر عدة مرات .

• تنفيذ الأوامر بسرعة كبيرة جداً .. وبالتالى يمكن تنفيذ أى جزء من البرنامج لأى عدد من المرات .. لذلك لابد من وجود شرط لإيقاف التكرار بعد عدد محدد من المرات .

6 – إخراج النتائج Output :

بعد الإنتهاء من إدخال البيانات وتحويلها إلى معلومات تظهر لنا مخرجات النتائج المطلوبة من خلال .. (الشاشة / الطابعة / الـ CD / DVD) .

• **والرسم التالى يوضح العمليات التى يقوم بها جهاز الكمبيوتر :**



س / ماهى البرمجة Programming ؟

البرمجة .. هى مجموعة من الخطوات المرتبة ترتيباً منطقياً والتى تستخدم لحل المشكلات (المسائل) ويطلق على هذا النوع أسم (الخوارزم Algorithm) .

• **فمثلاً** .. عندما نذهب يومياً إلى العمل فإننا نقوم بعمل مجموعة من الخطوات المرتبة والمبرمجة .. لذلك فإن الـ Algorithm هى خطة العمل لأى مشروع .



من الأخطاء التى يقع فيها بعض المبرمجين .. هو أن يبدأ فى إعداد البرنامج قبل أن يقوم بإعداد الخطوات المرتبة Algorithm .

س / كيف تتم كتابة الأكواد البرمجية ؟

تكتب الأكواد البرمجية باللغة الإنجليزية ، وتختلف قواعد البرمجة بإختلاف أنواع لغات البرمجة .

- وكل لغة برمجة تحتاج إلى برنامج خاص بها يسمى بالمفسر Interpreter ، والمترجم Compiler .

س / ما هو الفرق بين المفسر Interpreter ، والمترجم Compiler ؟

- **المفسر Interpreter :**

يقوم بتحويل اللغة عالية المستوى التى تمت الكتابة بها إلى لغة الآله .. بمعنى تحويل الأمر الأول ثم تنفيذه مباشرة .. ثم تنفيذ الأمر الثانى .. وهكذا حتى ينتهى البرنامج .

- **المميزات** .. سهولة إكتشاف الخطأ وإصلاحة .
- **العيوب** .. بطئ التشغيل .

- **المترجم Compiler :**

ويقوم المترجم بفحص الكود بالكامل وتصحيح أى خطأ موجود بالكود .. ثم يقوم بتحويل جميع لغة البرنامج إلى لغة الآله قبل البدء فى تنفيذ أى أمر .

- **المميزات** .. عملية التشغيل تكون أسرع من البرامج المكتوبة باللغات التى تعتمد على المفسر .
- **العيوب** .. صعوبة تصحيح الأخطاء .



تتميز لغة الـ Visual Basic بإستخدام المفسر Interpreter فى مرحلة الكتابة وتصحيح الأخطاء .. ثم بعد ذلك تقوم بترجمتة من خلال المترجم Compiler .. حتى تستفيد من مميزاته ، وسرعة تشغيل البرنامج .. وتلافى عيوب كل طريقة .

■ البرمجة المسيرة بالأحداث Event Driven Programming :

كنا فى الماضى نتعامل مع نظام تشغيل الـ DOS وهو عبارة عن شاشة سوداء يتم الكتابة عليها مجموعة من الأوامر .. كما بالشكل التالى :

```
 Volume in drive C is Win 7
 Volume Serial Number is 6263-6D90

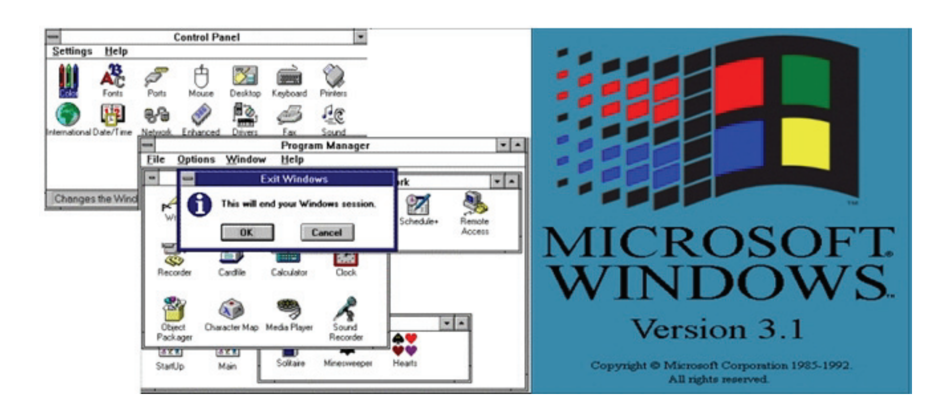
 Directory of C:\Users\LOLY

11/19/2013  05:31 AM    <DIR>          .
11/19/2013  05:31 AM    <DIR>          ..
11/19/2013  05:31 AM    <DIR>          Contacts
11/22/2013  05:02 AM    <DIR>          Desktop
11/20/2013  07:49 AM    <DIR>          Documents
11/19/2013  06:41 AM    <DIR>          Downloads
11/20/2013  01:53 AM    <DIR>          Favorites
11/19/2013  05:31 AM    <DIR>          Links
11/19/2013  05:31 AM    <DIR>          Music
11/19/2013  05:31 AM    <DIR>          Pictures
11/19/2013  05:31 AM    <DIR>          Saved Games
11/19/2013  05:31 AM    <DIR>          Searches
11/19/2013  05:31 AM    <DIR>          Videos
               0 File(s)              0 bytes
              13 Dir(s)  18,830,352,384 bytes free

C:\Users\LOLY>
```

حتى ظهر نظام التشغيل Windows فتغيرت هذه المفاهيم لتزيد من إستخدام الكمبيوتر والسهولة والمتعة .. من خلال التعامل مع الأيقونات Icons ، وإستخدم الـفأرة Mouse ، والبيئة الرسومية التى نراها أثناء التعامل مع

جهاز الكمبيوتر وكانت أول نسخة لنظام تشغيل Windows 3.1 .. كما بالشكل التالى :



• **إذن** .. نظام التشغيل Windows زاد من التفاعل بين المستخدم وجهاز الكمبيوتر من خلال التعامل مع الأيقونات والنوافذ وسحبها ووضعها فى مكان أخر .

• ومن هنا تغيرت طريقة عمل المبرمج التى تعمل تحت نظام التشغيل Windows ، وأيضاً تغير أسلوب لغات البرمجة المستخدم فى تصميم البرامج تحت نظام الـ Windows .. مثل تغيير نظام العمل فى لغة الـ Basic .. كما بالشكل التالى :

```
File  Edit  View  Search  Run  Debug  Calls  Options
                              Untitled
CLS
COLOR 11                    'Make it not look normal
WHILE INKEY$ <> "x"         'Exit when "x" is pressed
  r% = RND * TIMER
  r% = r% MOD 2             'This stuff randomly produces 0 or 1
  PRINT r%;                 'Then write it onto the screen
  'Optional:
  PRINT CHR$(7);            'This makes a continuous beep, we won't use this
WEND
END
```

لتتحول إلى لغة Visual Basic .. أى لغة البيسك المرئية كما بالشكل التالى :

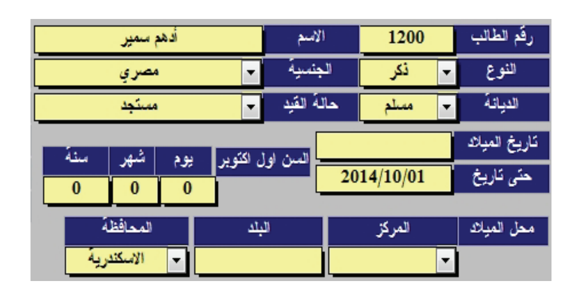


• لذلك .. نجد أن لغة الـ Visual Basic عبارة عن بيئة رسومية يكتب فيها البرنامج بإستخدام أسلوب يسمى **" البرمجة المسيرة بالأحداث "** .. وهو أسلوب يختلف عن أسلوب البرمجة المتبع فى كتابة البرامج القديمة تحت نظام تشغيل الـ DOS .

حيث كان البرنامج داخله عبارة عن مجموعة من الأوامر تكتب بشكل متتابع وتخزن فى ملف واحد .. ثم يقوم الكمبيوتر بتنفيذها واحده وراء الأخرى بنفس بنفس ترتيبها داخل البرنامج .

• أما فى نظام البرمجة المسيرة بالأحداث Event Driven Programming فينتظر البرنامج الحدث الذى يقوم به المستخدم .. مثل :

- الضغط على مفتاح .
- الإختيار من القائمة .
- وضع الـ Mouse فى مكان معين .



وعندها يقوم البرنامج بتنفيذ مجموعة من الأوامر كرد فعل على الحدث Event الذى قام به المستخدم .



على المبرمج الذى يقوم بكتابة برنامج بلغة الـ Visual Basic يجب وضع فى إعتباره جميع الأحداث التى من الممكن أن يقوم بها المستخدم .. ويكتب لكل حدث مجموعة من الأوامر والتعليمات اللازمة لها .

■ البرمجة بالكائنات Object Oriented Programming (OOP):

• البرنامج :

عبارة عن مجموعة أوامر تجمع وتخزن فى ملف واحد ، ويقوم جهاز الكمبيوتر بتنفيذها بالترتيب .. مثل الأوامر (Open / Copy / Repeat) ، وكان يطلق على هذا الأسلوب فى البرمجة أسم **Linear Programming** **البرمجة الخطية** .. وهو ما كانت تستخدمة لغة الـ Basic قديماً .

أما فى الإتجاهات الحديثة فى عالم البرمجة .. ظهرت عدة تقنيات حديثة مستخدمة فى ذلك .. مثل لغات (Visual Basic / Java / C++) .. ويسمى هذا الأسلوب بأسم **" البرمجة بالكائنات " OOP** .

- ومن هنا .. قامت فكرة الـ OOP على أساس المحاكاة Simulation للواقع أو العالم الحقيقى .. فكل ما يحيط بنا عبارة عن كائنات ، أو Object مثل الإنسان ، والجماد ، والطيور .. ولكن لكل مجموعة من الكائنات لها مجموعة من الخصائص .. مثل (لون العين للإنسان / شكل جناح الطائر) .

- ولكل Object .. مجموعة من الوظائف أو الأساليب الـ Methods التى يقوم بها .. مثل حركة الطائر فى أى إتجاه .

- وبتطبيق ذلك على لغة الـ Visual Basic .. نجد أن المفتاح Button له مجموعة من الخصائص ، والأساليب تتحدد أثناء الكتابة .

- كما يتعامل مع مجموعة من الأحداث .. مثل الضغط على المفتاح لتنفيذ الوظيفة أو تحريك الـ Mouse فو ق المفتاح .. أى أن الـ Object داخل لغة الـ Visual Basic مثل أى Object فى الحياة له 3 مفاهيم أساسية :

1 – الوراثة Inheritance :

ومنها يمكن إنشاء أى Object جديد .. من Object قديم يرث عنه صفاته وخصائصه .

2 – **الإحتوائية Encapsulation :**

وهو أن كل Object يحتوى بداخله على كل المعلومات ، أو البيانات التى تصفه .. وفى نفس الوقت يحتوى الـ Methods التى تتعامل مع البيانات Data بما يمكننا من فرض الحماية على هذه المعلومات Protection .

3 – **تعدد الأشكال Polymorphism :**

وتعنى أنه بالرغم من وجود نفس الـ Methods للعديد من الـ Objects إلا أن كل Object يؤدى الـ Method بطريقة مختلفة عن الأخر على حسب طبيعة هذا الـ Object .

- **فمثلاً** .. حركة الطيران للطائرة فى الجو لها Object ، وMethods مختلف عن حركة طيران أى طائر مثل العصفور .. فالطائرة تطير على إرتفاع عالى جداً ، أما العصفور فيطير على إرتفاع منخفض .

- **إذن :**

البرمجة داخل الـ Visual Basic .. بها مجموعة من الصفات Object ، والخصائص Properties التى يمكن التحكم فيها وتغييرها .. ولها مجموعة من الوظائف التى يجب على المبرمج إستخدامها فى تنفيذ برنامجة .

وأيضاً مجموعة من الأحداث Events التى تستجيب لها .. ويقوم المبرمج بمعالجة الـ Events المهمة التى يتوقع أن يقوم بها المستخدم .

■ مراحل إنشاء أى برنامج بإستخدام لغة الـ Visual Basic :

يمر أى برنامج بعدة مراحل .. وتسمى هذه المراحل عملية تطوير النظم System Developer .. وأهم هذه المراحل :

1 – التحليل Analysis :

• وفيها يقوم محلل نظم المعلومات ، أو المبرمج مع نفسه ومع برنامجه للإطلاع على جميع الأوراق والمستندات المتعلقة بالنظام .. ليصل فى النهاية إلى حل لأبعاد المشكلة المطلوب حلها بواسطة البرنامج أو النظام .

• ثم بعد ذلك يحدد المدخلات Input التى يتم إدخالها للبرنامج ، والمخرجات Output التى ستخرج منه .. وذلك إذا كان البرنامج كبيراً ومعقداً .. فنطلق عليه كلمة System أو نظام .

• أما لو كانت المشكلة بسيطة والهدف المطلوب تحقيقة من البرنامج بسيط فيكفى فقط أن يضع المبرمج يده عليها ليعرف ما هى المشكلة ؟ وما هو المطلوب من برنامجة ؟

2 – التصميم Design :

وفى هذه المرحلة يضع مصمم البرنامج أو الـ Designer الخطوط العريضة والهيكل الكامل للبرنامج على الورق .. ويحول البيانات والمعلومات التى قام بها محلل النظم إلى شكل يصلح صياغتة .. وبعد ذلك على هيئة أكواد برمجية تكتب بأى لغة من لغات البرمجة .



• فى هذه المرحلة يتم تحديد كل أنواع البيانات التى سيتعامل معها النظام ، وتحديد المدخلات ، والمخرجات ، وتحديد شكل الشاشات التى سيحتويها النظام وأيضاً الرسومات وخرائط التدفق Flow Chart التى توضح سير عمل البرنامج .

3 – البرمجة Programming :

يقوم المبرمج فى هذه المرحلة بتحويل كل ما سبق إلى أكواد برمجية يتم كتابتها بإحدى لغات البرمجة التى تناسب النظام الذى نقوم بإنشاؤه .. ويصاحب هذه المرحلة تشغيل دائم للبرنامج وتصحيح الأخطاء إن وجدت .

4 – إختبار البرنامج Testing :

- يتم إختبار البرنامج .. وذلك بإعطاؤه بيانات حقيقية لنرى هل بالفعل سوف يؤدى المطلوب منه بنجاح أم لا ؟ وهل يوجد خطأ معين ؟

- وتستمر هذه المرحلة حتى يتأكد المبرمج تماماً بأن البرنامج سوف يؤدى المطلوب منه بنجاح .

- ثم بعد ذلك يتم توزيع البرنامج على المستخدمين وتدريبهم عليه إذا كان النظام المستخدم معقداً .. وبعد ذلك تستمر العلاقة بين المبرمج والبرنامج فى شكل صيانة للنظام System Maintenance .

الفصل الثانى

التعرف على بيئة الـ Visual Basic



- خطوات تحميل نسخة برنامج .. Microsoft Visual Studio 2013

- خطوات إعداد البرنامج على جهاز الكمبيوتر .

- خطوات فتح وتشغيل برنامج .. Microsoft Visual Basic 2013

- التعرف على واجهة إستخدام برنامج .. Visual Basic 2013

- التعرف على النافذة Form .
- التعرف على صندوق الأدوات Tool Box .
- نافذة الخصائص Properties .
- نافذة الكود Code Editor .

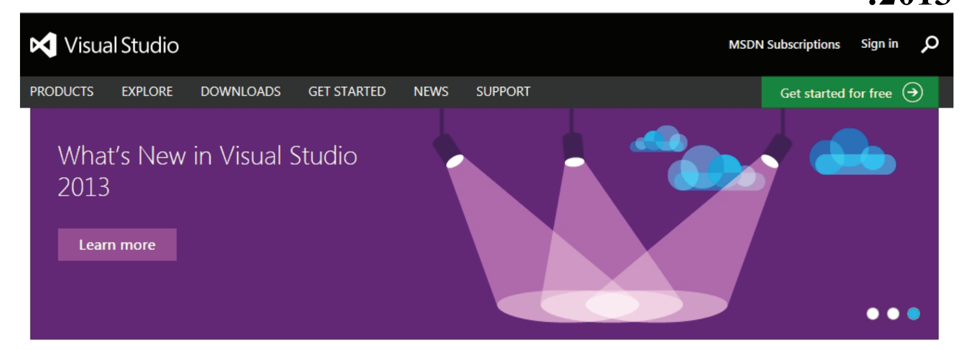
الفصل الثانى

التعرف على بيئة الـ Visual Basic

فى البداية كلمة **BASIC** هى إختصار للجملة ..
Beginner's **A**ll-Purposes **S**ymbolic **I**nstruction **C**ode
أى شفرة تعليمات رمزية لجميع أغراض المبتدئين .. وكلمة Visual تعنى مرئى أى أن Visual Basic لغة البيسك المرئية .

ولغة الـ Visual Basic هى إحدى منتجات شركة Microsoft العالمية ضمن مجموعة Microsoft Visual Studio .. ولتحميل النسخة التجريبية من البرنامج من خلال الموقع الرسمى لشركة ميكروسوفت :
www.microsoft.com

- **خطوات تحميل نسخة برنامج Microsoft Visual Studio 2013:**



Any App, Any Team

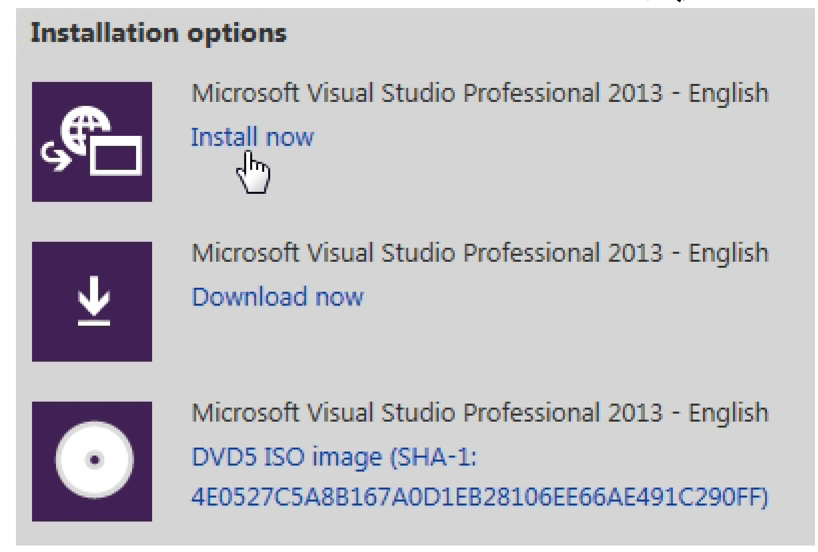
Visual Studio is a comprehensive collection of tools and services to help you create a wide variety of apps, both for the Microsoft platform and beyond. Visual Studio also connects all of your projects, teams, and stakeholders. Now your team can work with greater agility from virtually anywhere—irrespective of development tool, including Eclipse and Xcode. Whether you're designing mission-critical .NET apps, writing blazing fast code with C++ AMP, or testing and debugging a cloud-connected HTML/JavaScript app that runs on many devices, join millions of developers worldwide in choosing Visual Studio as your essential development environment.



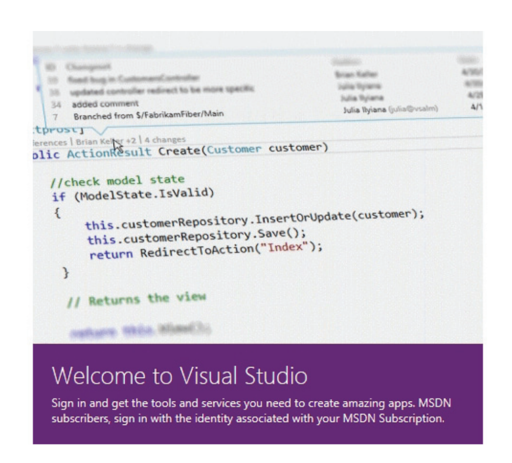
1 – إضغط على الرابط Download من موقع شركة ميكروسوفت .. فتظهر النافذة التالية :

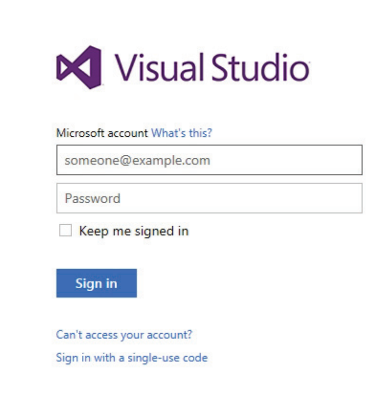


2 – إضغط على الأمر Visual Studio Professional 2013 .. فتظهر النافذة التالية :



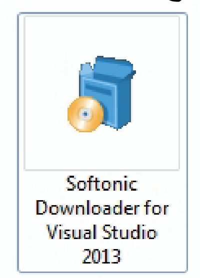
3 – إضغط على الأمر Install Now .. فتظهر الصفحة التالية الخاصة بإدخال أسم المستخدم وكلمة المرور الخاصة بك على موقع شركة ميكروسوفت :





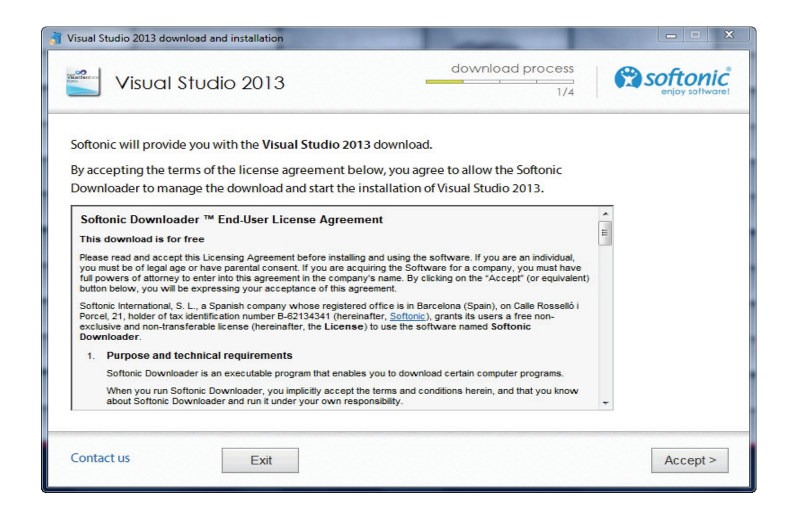
Don't have a Microsoft account? Sign up now

4 – يقوم الموقع بتحميل البرنامج التالى :

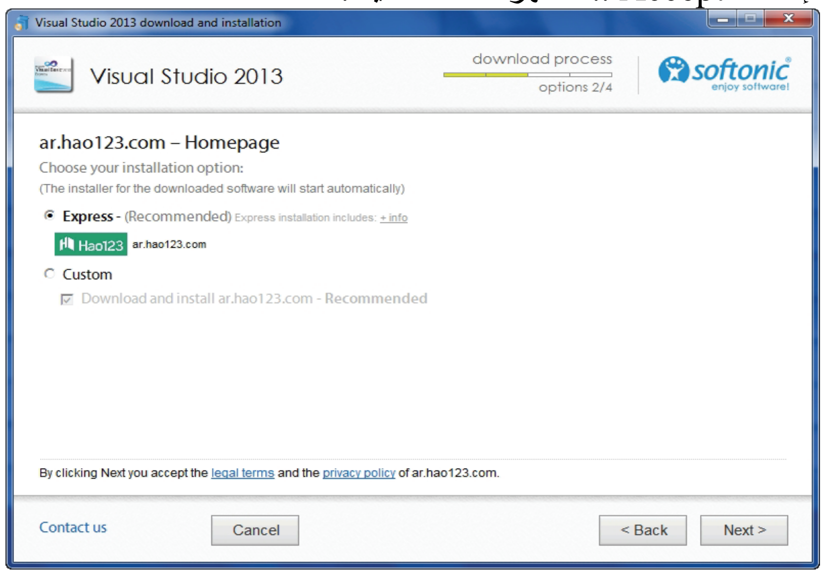


وهو البرنامج الخاص بتحميل نسخة Microsoft Visual Studio 2013

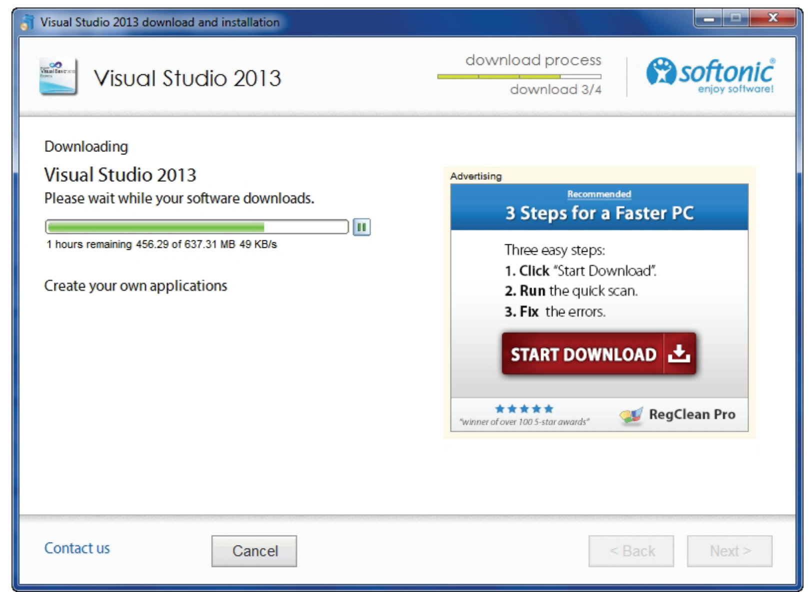
5 – إضغط على الأيقونة السابقة .. فتظهر النافذة التالية :



6 – إضغط Accept .. فتظهر النافذة التالية :

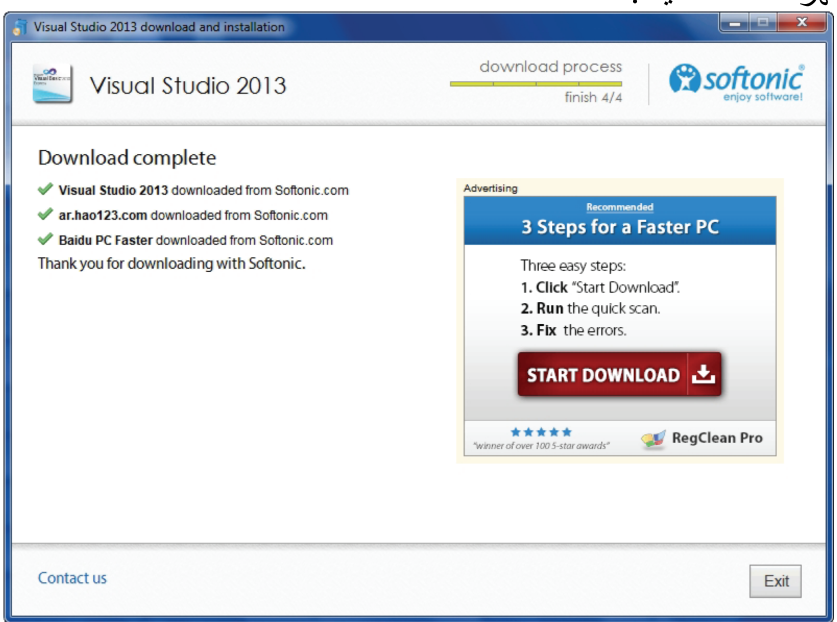


7 – نشط الإختيار Express .. ثم إضغط Next ، ثم نشط الإختيار Express مرة أخرى .. وإضغط Next مرة أخرى فتظهر النافذة التالية :

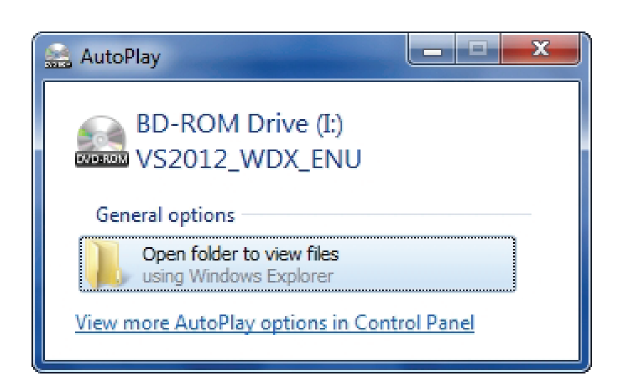


8 – إنتظر مدة التحميل حتى ينتهى البرنامج من تحميل نسخة برنامج :
Microsoft Visual Studio 2013

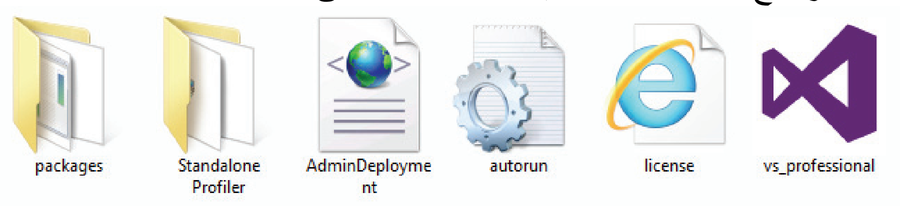
فتظهر النافذة التالية :



9 – إضغط Exit .. وذلك لإنهاء تحميل نسخة البرنامج .. فتظهر النافذة التالية :



10 – إضغط على الأمر Open Folder to View File .. فتظهر أيقونات البرنامج داخل مجلد التحميل كما بالشكل التالى :



■ خطوات إعدادات البرنامج على جهاز الكمبيوتر :

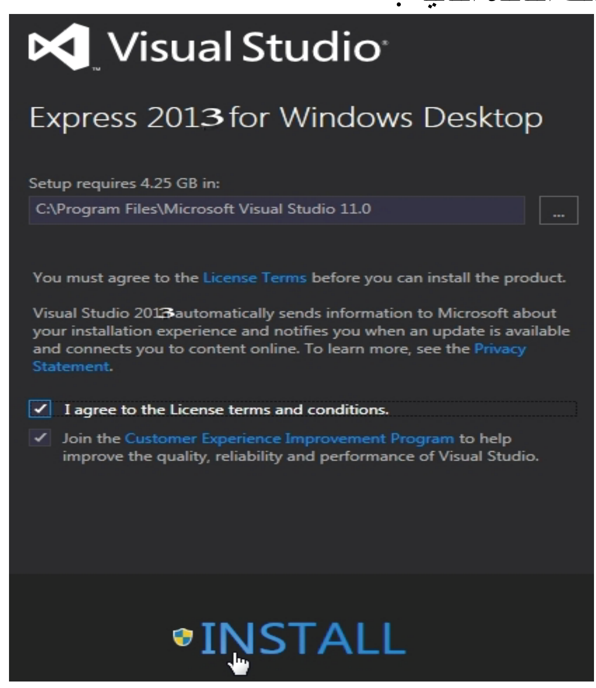
Microsoft Visual Studio 2013 Setup

1 – إضغط على الأيقونة 

VS-Professional من مجلد البرنامج فتظهر النافذة التالية :



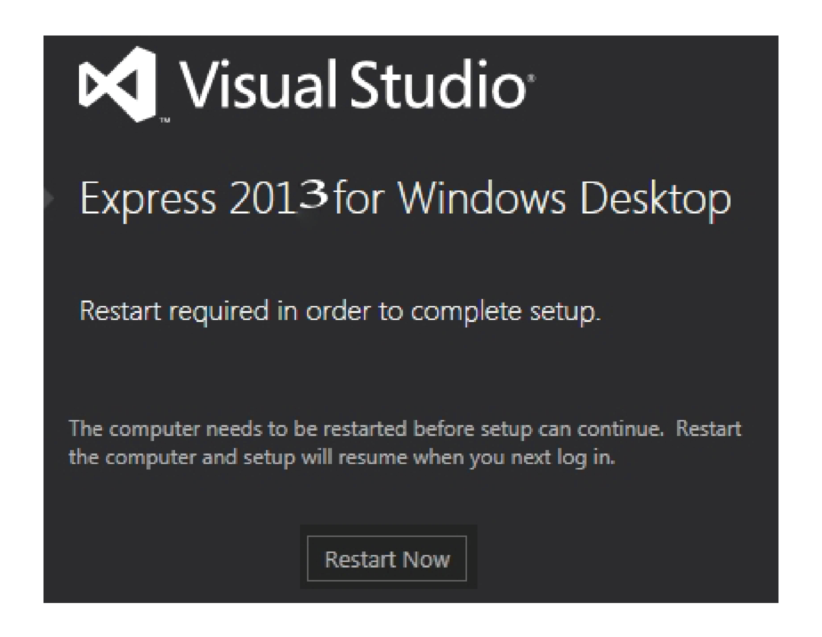
ثم تظهر بعد ذلك النافذة التالية :



2 – نشط الإختيار I Agree to the License Terms and Condition
.. ثم إضغط Install .. فتظهر النافذة التالية :



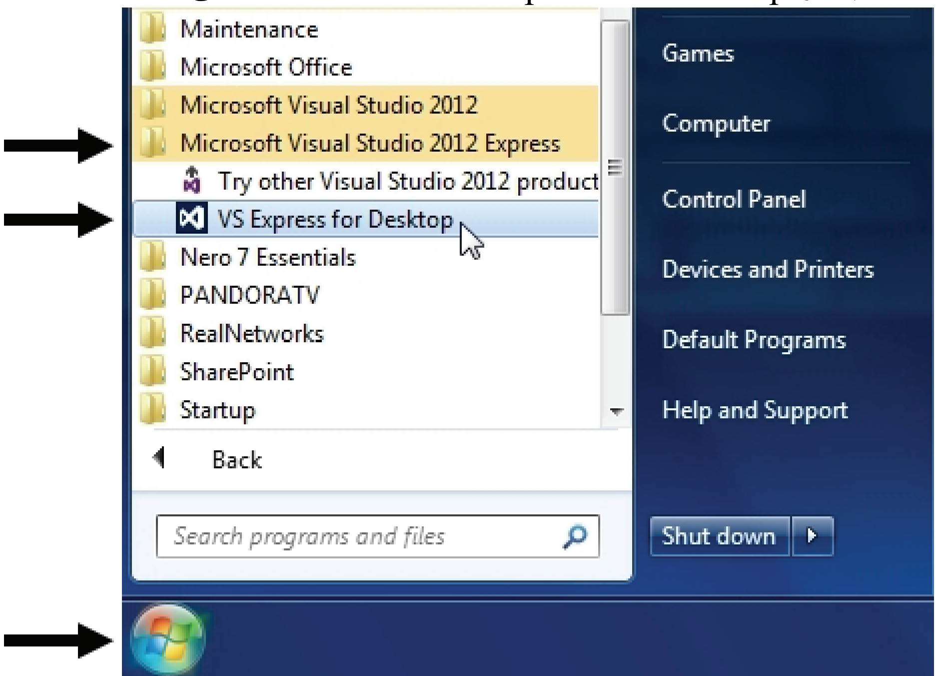
إنتظر قليلاً حتى يتم الإنتهاء من إعدادات البرنامج .. فتظهر النافذة التالية :



3 – إضغط Restart Now .. فيتم إعادة تشغيل جهاز الكمبيوتر مرة أخرى .

■ خطوات فتح وتشغيل برنامج Microsoft Visual Basic 2013 :

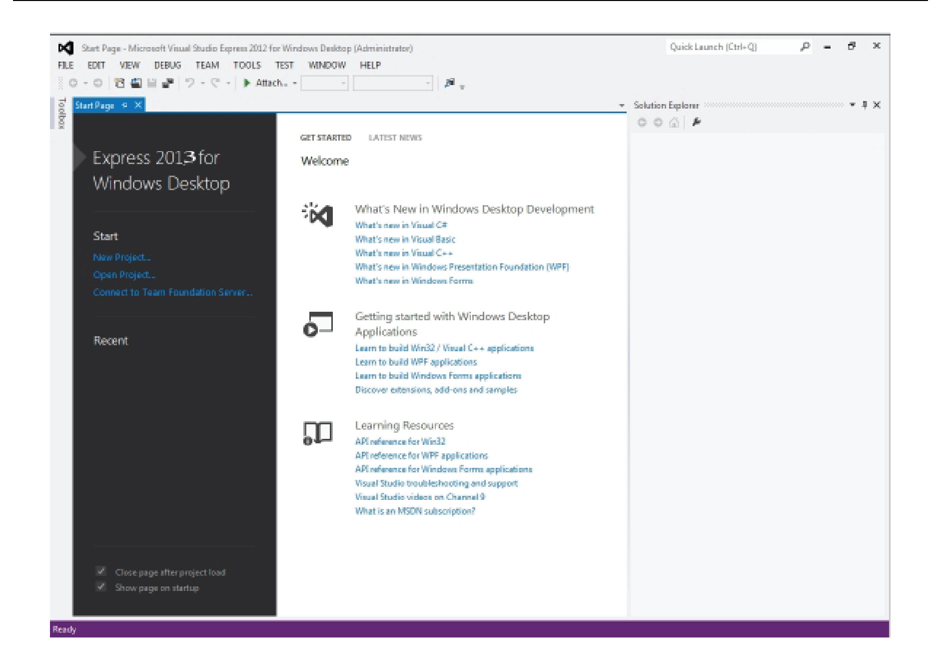
1 – من القائمة Start .. إضغط All Programs .. ثم إضغط على الأمر Microsoft Visual Studio 2013 Express .. فتنسدل منها قائمة إختار VS Express for Desktop .. كما بالشكل التالى :



أو عمل أيقونة مختصرة على سطح المكتب Desktop ..



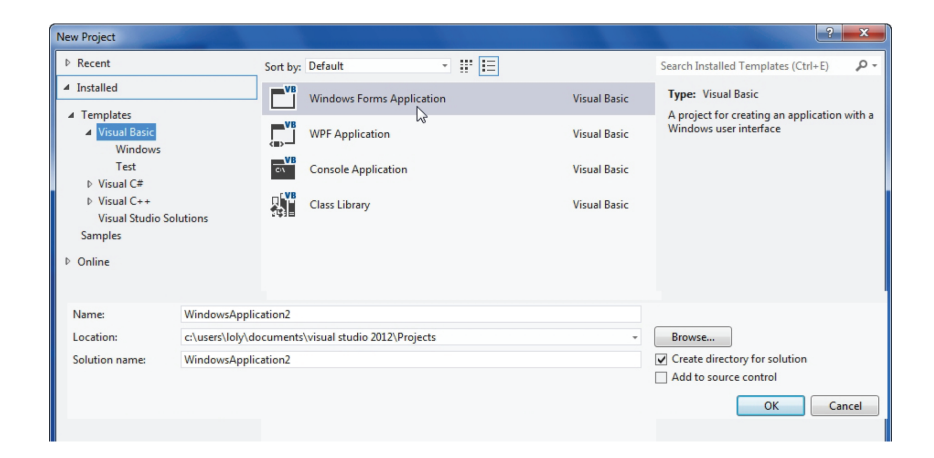
2 – تظهر النافذة الرئيسية لبرنامج Microsoft Visual Studio 2013 كما بالشكل التالى :



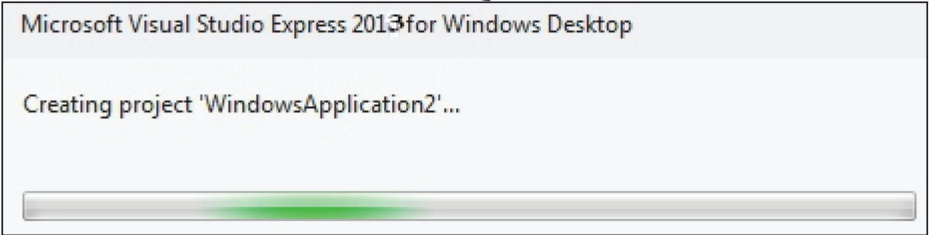
3 – إضغط على الأمر New Project من الجزء Start يسار نافذة البرنامج ..

Start
New Project...
Open Project...
Connect to Team Foundation Server...

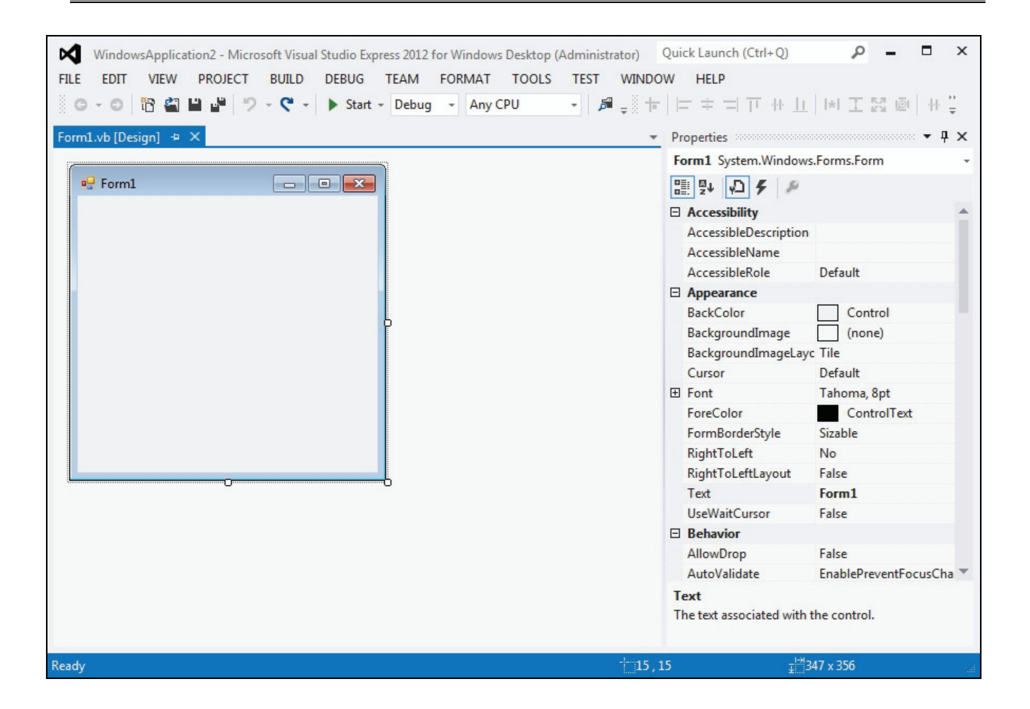
فتظهر النافذة التالية :



4 – من النافذة السابقة إضغط على Visual Basic يسار نافذة البرنامج ، ثم إضغط على الأمر Windows Form Application .. ثم إضغط **OK** فتظهر نافذة تحميل البرنامج التالية :



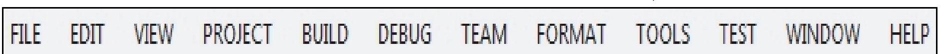
- إنتظر قليلاً حتى تظهر النافذة الرئيسية لبرنامج Visual Basic .. كما بالشكل التالى :



■ التعرف على واجهة إستخدام برنامج Visual Basic 2013 :

تتكون نافذة الـ Visual Basic 2013 من الأجزاء التالية :

■ أولاً : شريط القوائم Menu Bar :



تحتوى نافذة البرنامج على 12 قائمة متغيرة المحتوى .. وكل قائمة تحتوى على مجموعة من الأوامر تظهر بالضغط على أسم القائمة .. وتستخدم هذه القوائم والأوامر التى بداخلها للتعامل مع الـ Visual Basic والتحكم فى وظائفها .. لكى نتمكن من إنشاء التطبيقات Application المطلوبة .



تختص كل قائمة كما هو متعارف عليه فى جميع البرامج بمجموعة من الأوامر التى لها علاقة ببعضها .. " وهو ما يتم تصميمة لاحقاً " .

• **القائمة File :**

FILE	
New Project...	Ctrl+Shift+N
New Team Project...	
New File...	Ctrl+N
Open Project...	Ctrl+Shift+O
Open File...	Ctrl+O

وبها أهم الأوامر الأساسية مثل :
- إنشاء مشروع جديد New Project .
- فتح مشروع سابق Open Project .
- حفظ المشروع Save Project .

وغيرها من الأوامر الأساسية .

• **القائمة Edit :** وتستخدم عند كتابة كود البرنامج .

• **القائمة View :** تستخدم لإظهار وإخفاء عناصر الفيجوال بيسك .

• **القائمة Project :**
تستخدم للتعامل مع عناصر المشروع الواحد بالإضافة أو الحذف .

• **القائمة Format :**
تستخدم للتحكم فى تنسيق العناصر الموضوعة على النموذج Form .

• **القائمة Debug :**
وهى مسئولة عن إختبار البرنامج وإكتشاف الأخطاء وتصحيحها .

• **القائمة Run :** تستخدم لتشغيل وإيقاف البرنامج .

- وهكذا لباقى قوائم البرنامج .. التى سيتم تناول أوامر منها أثناء تنفيذ المشروع والتطبيقات لاحقاً .

■ **ثانياً : شريط الأدوات القياسى Standard Bar :**



وهو موجود أسفل شريط القوائم وبه مجموعة من المفاتيح المختصرة والتى تؤدى نفس الوظائف الموجودة فى القوائم .. ولكن تم وضعها فى شكل شريط للأدوات لتأدية الوظائف بسهولة وسرعة .

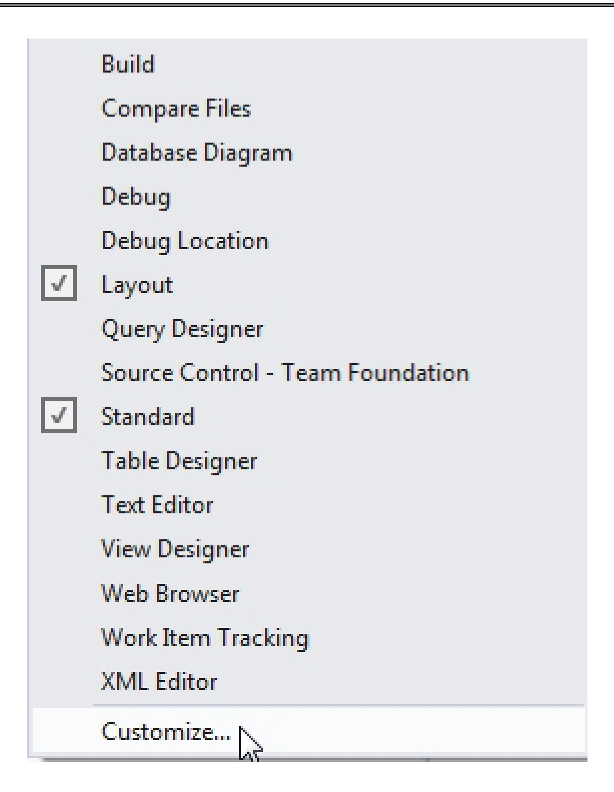


عند الوقوف بمؤشر الـ Mouse عند أى مفتاح فى شريط الأدوات القياسى يظهر تلميح Screen Tip به وظيفة هذا المفتاح .. كما بالشكل التالى :

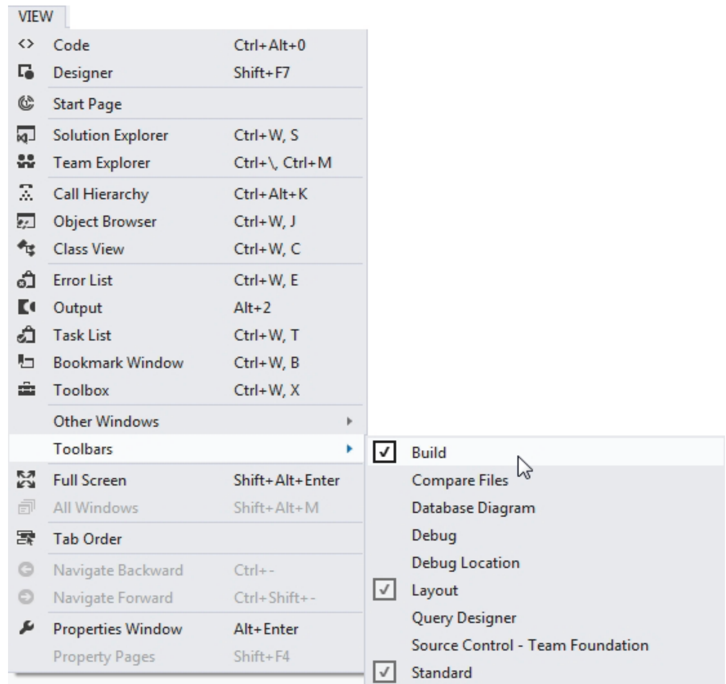




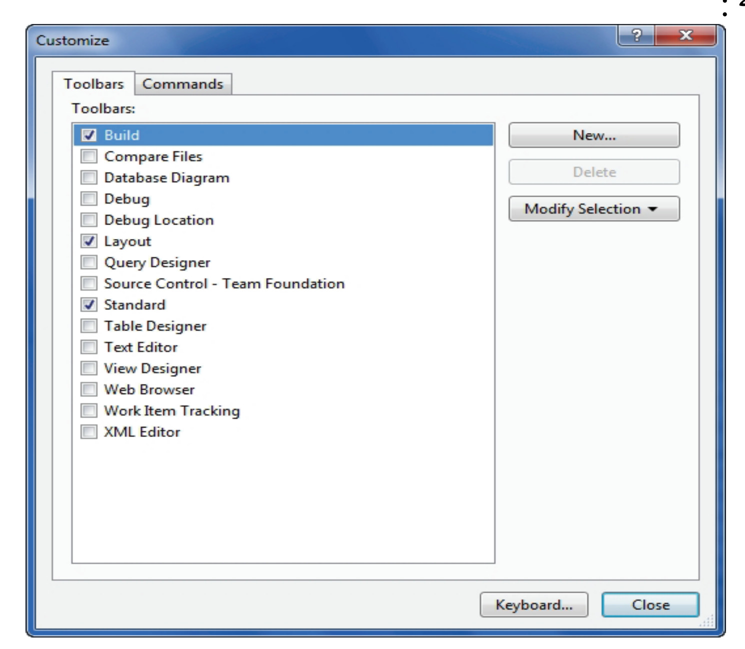
يمكن إضافة أشرطة أدوات أخرى لنافذة البرنامج .. وذلك بالضغط بمفتاح الـ Mouse الأيمن على هذا الشريط .. فتظهر قائمة مختصرة وبها مجموعة أوامر .. كما بالشكل التالى :



- أو من القائمة View إختار الأمر Tool Bars .. ثم إختار أى أمر تريده من القائمة الفرعية .. كما بالشكل التالى :



- كما يمكن تخصيص أى Tool Bars جديدة .. وذلك بالضغط على الأمر Customize من القائمة المختصرة أو الفرعية السابقة .. فتظهر النافذة التالية :



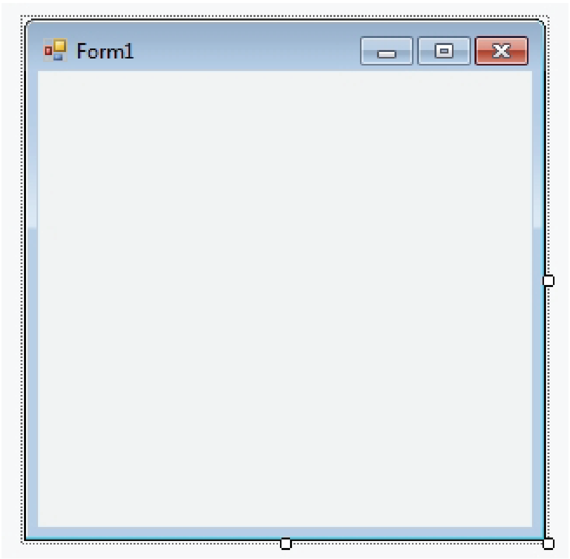
- نشط أى إختيار تريده .. ثم إضغط Close .



يجب معرفة أن بيئة تطوير الـ Visual Basic تعمل تحت نظام تشغيل Windows .. لذلك فمن المنطقى أن يكون العنصر الأساسى فيها هو النوافذ ومن المنطقى أيضاً أن تتكون التطبيقات Application التى نقوم بتنفيذها من أكثر من نافذة ، أو نافذة واحدة على الأقل .. وهى ما تسمى بالنماذج Forms .

■ **ثالثاً : النافذة Form :**

وهى تظهر فى منتصف الشاشة الرئيسية لبيئة تطوير الـ Visual Basic ..



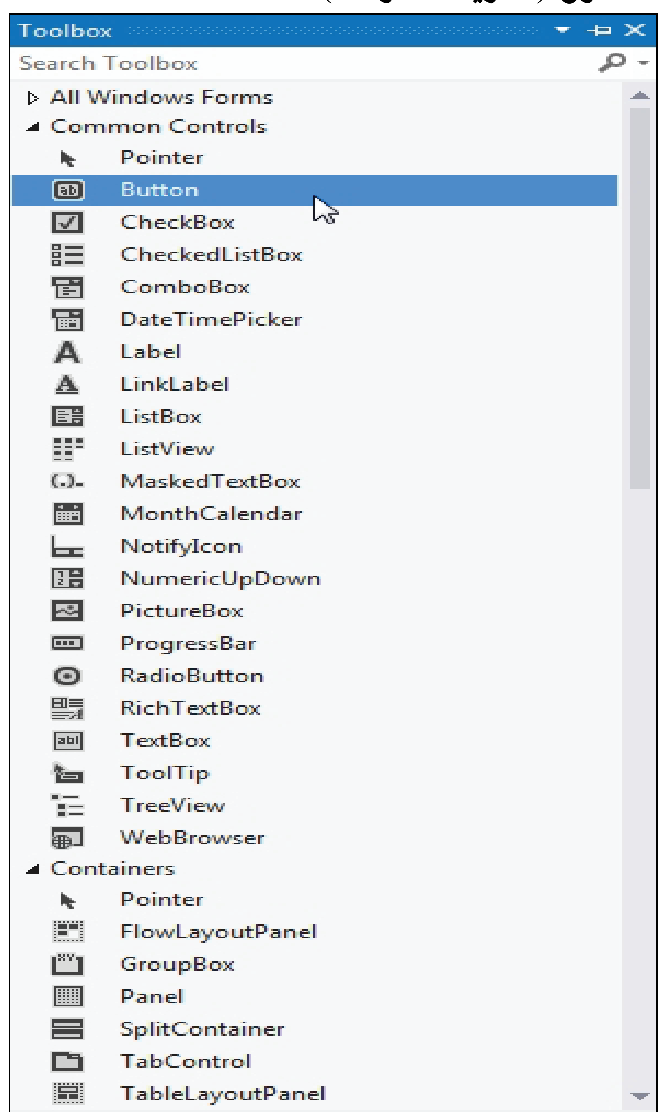
- وتستخدم النافذة Form Design لتصميم واجهة تطبيق البرنامج ويطلق عليها أسم Application Interface .

- ويظهر على نافذة النماذج أسم Form 1 وهو الأسم الإفتراضى للنموذج الأول وتوضع عليه (المفاتيح / مربعات النصوص / الصور) وغيرها من الأدوات الهامة المستخدمة فى تصميم واجهة البرنامج .

- ويستخدم النموذج Form فى العديد من التطبيقات التى نراها فى البرامج التى نعرفها ونتعامل معها.. مثل نوافذ (رسائل التحذير Dialog Box) .

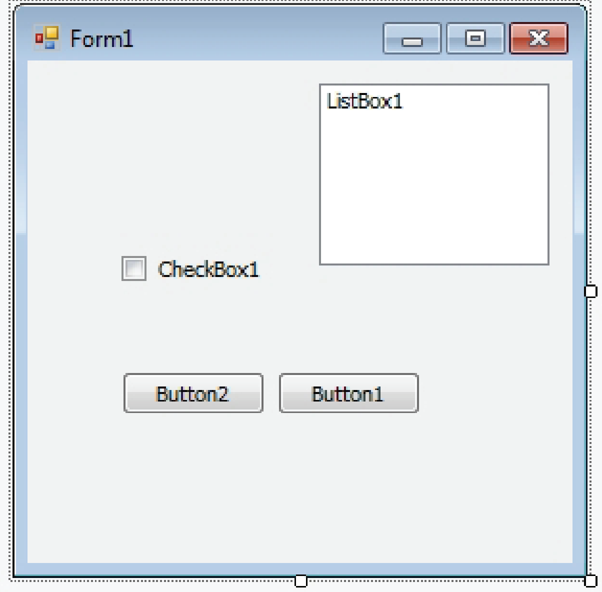
- كما يوجد للنموذج Form عدد 3 مقابض للتحكم فى حجم الـ Form أو النافذة بالتكبير أو التصغير رأسياً أو أفقياً أو قطرياً .

■ رابعاً : صندوق (شريط الأدوات) Tool Box :



• يستخدم هذا الشريط بالأدوات المختلفة بداخلة فى تشكيل واجهة التطبيق .

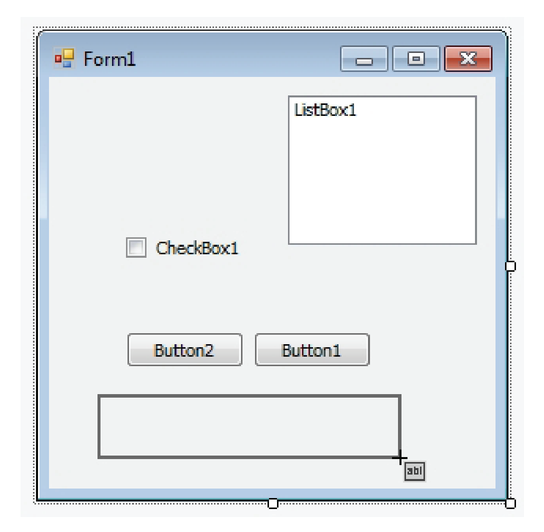
• ولإضافة أى Control أو أى عنصر من الـ Tool Box .. إضغط على العنصر المطلوب مرتين فيظهر داخل نافذة النموذج Form .. كما بالشكل التالى :



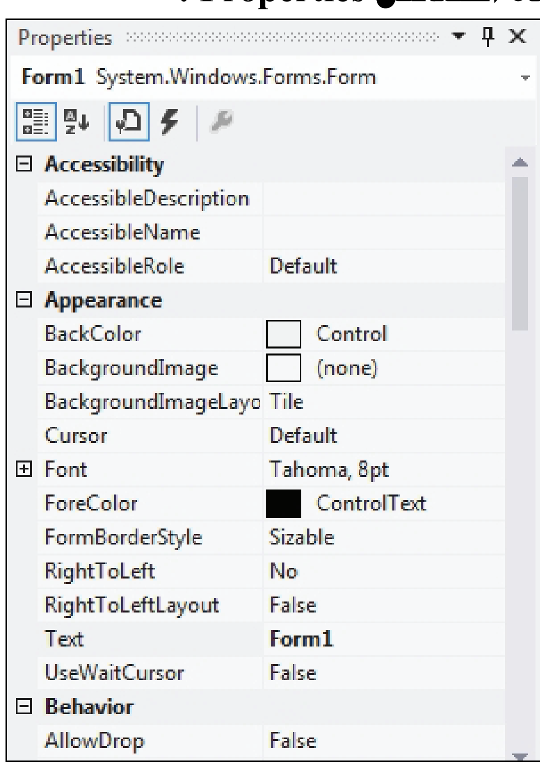
ثم نقوم بعد ذلك بضبط وتغيير مكان الـ Control أو العنصر على النموذج Form .. وذلك بالضغط والسحب بالـ Mouse فى المكان المناسب .



يمكن الضغط مرة واحدة على الـ Control ، أو العنصر المطلوب من صندوق الأودات .. ثم رسم العنصر بالشكل المناسب داخل النموذج Form .. كما بالشكل التالى :



■ **خامساً : نافذة الخصائص Properties :**

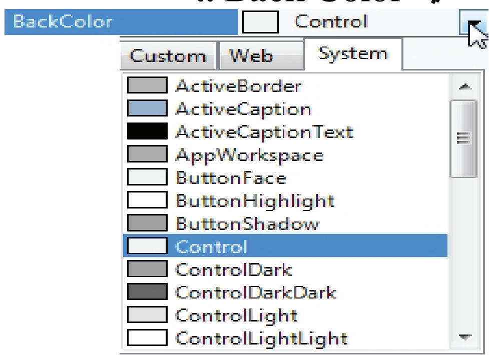


وهى نافذة خصائص العناصر التى تظهر على النموذج Form .. وفى حالة عدم ظهور هذه النافذة يتم الضغط على مفتاح **F4** من لوحة المفاتيح .. وتمكننا هذه النافذة من التحكم فى خصائص وعناصر واجهة التطبيق من الـ Form والـ Controls .. وتظهر الخصائص بها مرتبة ترتيب أبجدى .

- ولتغيير قيمة أى خاصية من خواص كل عنصر .. نقوم بإختيار الخاصية نفسها ثم تغيير قيمتها من القيمة المقابلة من العمود الأيمن .



- **هناك خصائص يتم تغيير قيمتها من خلال القائمة المنسدلة الى بجانب الإختيار مثل الخاصية Back Color ..**



- **وهناك خصائص أيضاً يتم كتابة قيمتها يدوياً .. مثل الخاصية Text ..**

Text	Form1

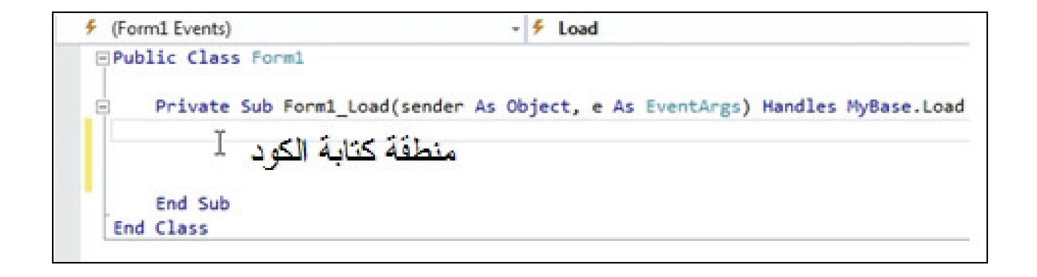
- **وهناك خصائص لا يمكن التحكم فيها إلا من خلال الكود البرمجى داخل البرنامج .**

- **تظهر أسفل نافذة الخصائص Properties ملاحظة توضح وظيفة الخاصية المختارة ..**

BackColor
The background color of the component.

■ **سادساً : نافذة الكود Code Editor :**

وتظهر بالضغط مرتين على نافذة النموذج Form .. أو بالضغط على العنصر المطلوب كتابة كود له .. فتظهر كما بالشكل التالى :





- **هناك نوافذ أخرى فى بيئة تطوير الـ Visual Basic يمكن إظهارها من القائمة View .**

- **إذن .. بيئة تطوير الـ Visual Basic تتميز بالمرونة والسهولة ونستطيع التحكم فيها بما يناسب إحتياجاتنا .. حتى نستطيع إخراج البرنامج بالشكل والكفاءة المطلوبة .**

الفصل الثالث
كتابة الكود داخل لغة
الـ Visual Basic

نتعلم فى هذا الفصل ..

- التعرف على ما هو الكود البرمجى Code ؟
- التعرف على القائمة المنسدلة Object.
- قائمة الإجراءات Procedure .

العمليات الحسابية داخل لغة الـ Visual Basic .

الفصل الثالث

كتابة الكود داخل لغة

الـ Visual Basic

لكى نتعلم كيفية كتابة الكود داخل لغة الـ Visual Basic .. لابد أن نتعلم ما هو الكود ؟

■ الكود Code :

هو عبارة عن أوامر يقوم المبرمج بتنظيمها وترتيبها بالشكل المناسب لكى تؤدى وظيفة محددة .. وهى نفسها الوظيفة التى صنع البرنامج من أجلها .

لذلك يجب أن نتعرف على المكان الذى سنقوم بكتابة الكود بداخله .. وهذا المكان .. هو نافذة تسمى بالـ Code Editor .

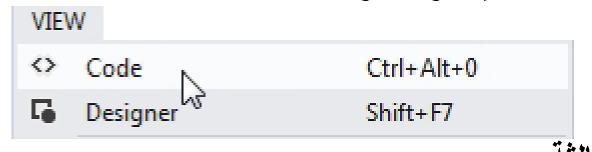
ولفتح نافذة الكود .. هناك أكثر من طريقة منها :

• **الطريقة الأولى :**

إضغط مرتين على أى عنصر موجود على الـ Form ، أو بالضغط على الـ Form نفسه مرتين .

• **الطريقة الثانية :**

من القائمة View إختار الأمر Code .

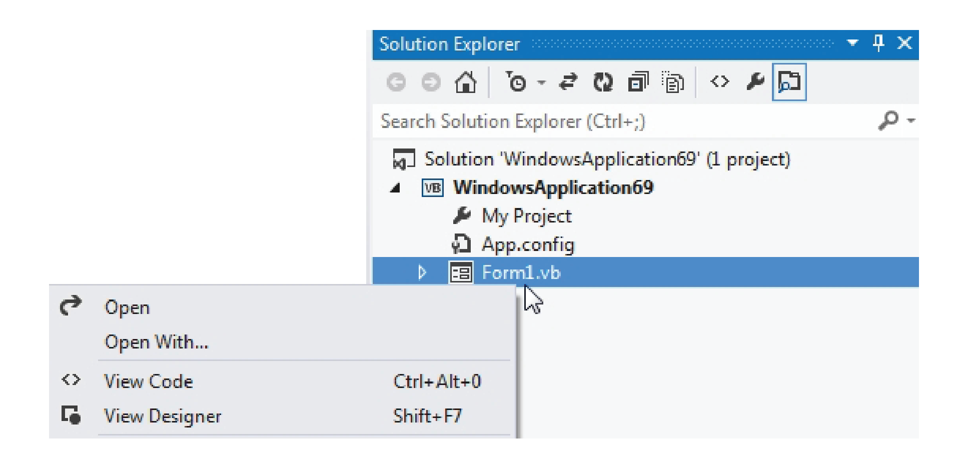


• **الطريقة الثالثة :**

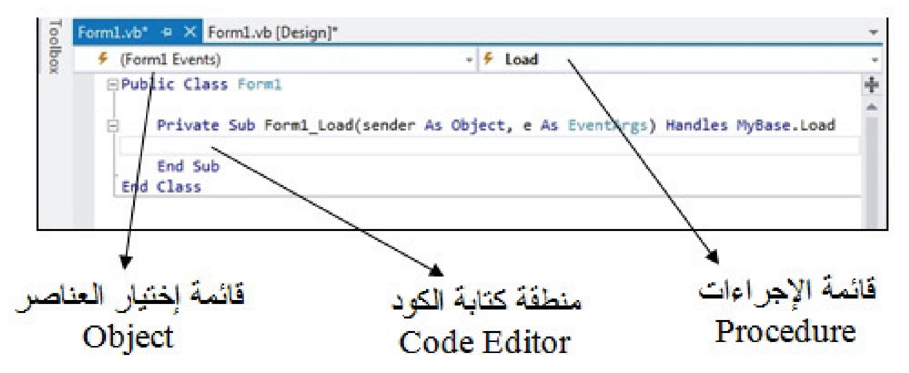
الضغط على مفاتيح **Alt + Ctrl + 0** من لوحة المفاتيح .

• **الطريقة الرابعة :**

الضغط بمفتاح الـ Mouse الأيمن على أسم النموذج Form من النافذة Solution Explorer .. فتظهر قائمة مختصرة .. إختار منها View Code ..

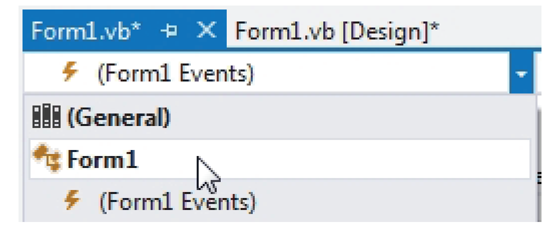


وبإختيار أحد الطرق الأربعة السابقة .. تظهر نافذة الكود التالية :



ومن النافذة السابقة نجد أن :

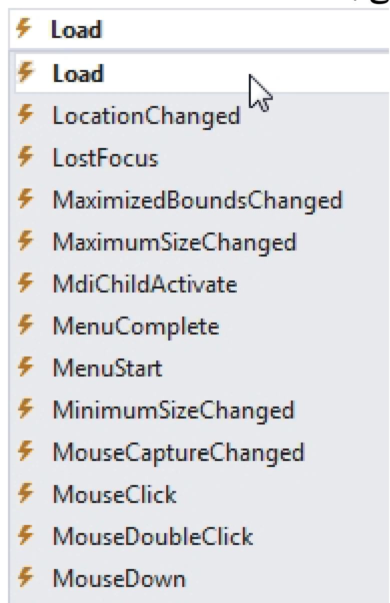
■ **القائمة Object :**



وتحتوى على جميع العناصر المضافة للبرنامج .. مثل مفاتيح التحكم ، وتظهر أسماء العناصر بالأسم الموجود فى الخاصية Text لكل عنصر .

■ **القائمة Procedure :**

وهى قائمة منسدلة بها جميع الإجراءات التى يحتويها البرنامج الذى يتم تصميمه .. وهى قائمة تظهر جميع الأحداث Events للعنصر المختار من القائمة Object ، وبإختيار أى إجراء منها .. يظهر مؤشر الـ Mouse داخل منطقة الكتابة الخاصة بالإجراء أو العنصر .. وهى موجودة أعلى يمين نافذة الكود .. كما بالشكل التالى :



• **الإجراء Procedure :**

هو مجموعة من الأوامر ، أو الجمل داخل لغة الـ Visual Basic .. والتى تؤدى وظيفة معينة يتم تجميعها ووضعها داخل مكان واحد .. هو الإجراء Procedure نفسه .

والإجراء يكون له أسم .. ويتم تنفيذ هذه الأوامر عند الحاجة إليها بإستدعاء هذا الإجراء بواسطة أسمه مهما كان عدد مرات هذا الإستدعاء .

• **والإجراء Procedure فى لغة الـ Visual Basic نوعان :**

1 – دالة Function .. تؤدى وظيفة ، ويقوم بإرجاع قيمة للنظام .

2 – **Sub Routines ..** يؤدى وظائف محددة .. ويكون الشكل العام كما بالشكل التالى :

```
Private Sub Form1_MenuStart(sender As Object, e As EventArgs) Handles Me.MenuStart

    End Sub
End Class
```



- **عندما يتغير محتوى القائمتين Procedure ، و Object يتغير مكان مؤشر الـ Mouse داخل نافذة الكود .. عند مكان العنصر المطلوب كتابة كود له .**

- **من نافذة الكود السابقة نجد أن الأجراء إما أن يكون :**

1 – خاص Private :

وهو أنه لم يمكنه إستدعاء هذا الـ Procedure إلا من الملف ، أو المكان الذى تم إنشاؤه فيه .

2 – عام Public :

وهذا معناه أن ننا نستدعى الـ Procedure من أى مكان فى البرنامج فى أى لحظة .. وعند كتابة كلمة public تظهر باللون الأزرق بمجرد ترك مسافة بعد كتابة الكلمة .

- **إذن :**

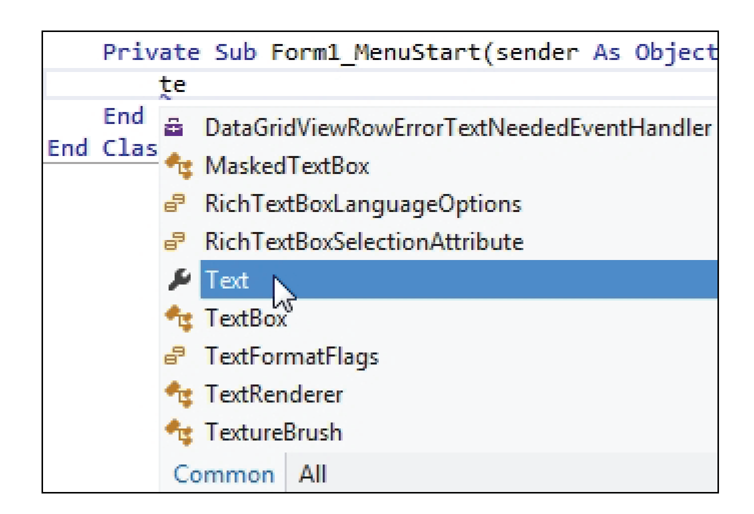
يتضح لنا مما سبق أنه من الممكن إنشاء الـ Procedure داخل ملف النموذج Form ، وتعريفه على أنه public .. بحيث يراه جميع أجزاء البرنامج .

س / ما هو الـ Code Editor ؟

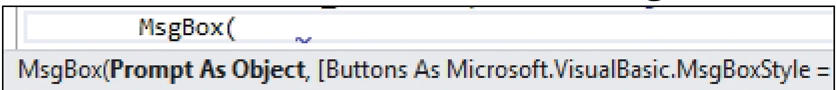
يظهر الـ Code Editor كمحرر نصوص عادى مثل برنامج Word ولكن فى الحقيقة هو ليس كذلك .. لأنه محرر نصوص خاص جداً به ذكاء إصطناعى ويستخدم فقط لكتابة كود لغة الـ Visual Basic .

كما يحتوى الـ Code Editor بداخله على جميع الأوامر والجمل المسموح للمستخدم بكتابتها .. لذلك فهو يحمل بعض الخصائص التى تساعد المستخدم فى كتابة الكود .

• **فمثلاً :**
لو تم وضع مؤشر الكتابة فى الجزء الخاص بسطر كتابة الكود فتظهر قائمة منسدلة تلقائية بها جميع الأوامر المرتبطة بالجملة المكتوبة .. ويمكن إختيار أياً منها .. كما بالشكل التالى :



• كما تسهل علينا أيضاً نافذة كتابة الكود .. ظهور شـــريط صيغة كتابة الأوامــر Toll Tip .. **فمثلاً** .. عند كتابة الكود الخاص بالرسائل يظهر .. كما بالشكل التالى :



• بمجرد فتح قوس لكتابة الجملة .. يظهر شريط وبه باقى صيغة الجملة البرمجية أو الكود .. وذلك للمساعدة على كتابة الكود بشكل صحيح .



يتميز الـ Code Editor داخل لغة الـ Visual Basic بإمكانية نسخ ولصق وقص الأكواد عن طريق الأوامر Copy ، و Cut ، و Paste .. بدلاً من كتابتها .

■ العمليات الحسابية داخل لغة الـ Visual Basic :

لإجراء أى عملية حسابية داخل لغة الـ Visual Basic مثل أى عملية حسابية داخل أى لغة برمجة ، أو البرامج الجاهزة .. فالرموز الرياضية ، والمنطقية متشابهه .. مثل :

• الرموز الرياضية :

الرمز	الوظيفة
=	Equal يساوى
+	Addition الجمع
−	Subtract الطرح
*	Multiply الضرب
/	Divide القسمة
^	Exponent الأس

• الرموز المنطقية :

الرمز	الوظيفة
>	Greater Than أكبر من
<	Less Than أقل من
>=	Greater Than or Equal to أكبر من أو يساوى

<=	أقل من أو يساوى Less Than or Equal to
< >	لا يساوى Not Equal to

• أولويات تنفيذ العمليات الحسابية :

1 – الأقواس الداخلية أولاً ، ثم الأقواس الخارجية للعملية الحسابية .
2 – عمليات الأسس .
3 – عمليات الضرب ، أو القسمة .. أيهما أقرب من اليسار لليمين .
4 – عمليات الجمع ، أو الطرح .. أيهما أقرب من اليسار لليمين .

وسوف يتم تناول العمليات الحسابية ، والخطوات السابقة فى الأمثلة والتطبيقات فى الفصول القادمة .

الفصل الرابع

التعامل مع المشروع Project

نتعلم فى هذا الفصل ..

- ما هو المشروع Project ؟
- إنشاء مشروع جديد New Project .
- إضافة نموذج جديد للمشروع الحالى Add Form .
- إضافة ملف مخزن سابقاً للمشروع الحالى .
- فتح مشروع موجود سابقاً ًستكمال العمل عليه أو تعديله
- حفظ ملفات المشروع الداخلية .
- تصميم واجهة التطبيق Interface .
- التعرف على النماذج Forms .
- التعرف على لوحة الخصائص Properties .
- الأساليب Methods .
- الأحداث Events .

الفصل الرابع

التعامل مع الـمشروع Project

■ المشروع Project :

- هو ما نقوم بإنشاؤه بإستخدام لغة الـ Visual Basic .. فهو البرنامج ولكن فى مرحلة التصميم .. ويتكون المشروع Project من عدة ملفات وعناصر ، وأول هذه الملفات هو الملف الرئيسى المسمى Project 1 .

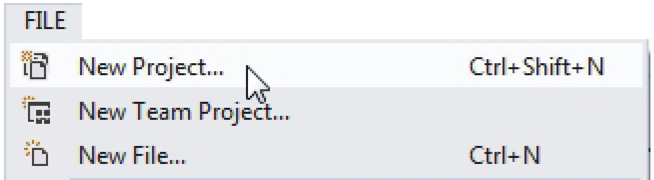
- ويحتوى الـ Project 1 على خصائص وصفات المشروع ويأخذ الملف الإمتداد VBA .

- بالإضافة إلى ملفات النماذج Forms وتأخذ الإمتداد Frm وتستخدم كنوافذ فى البرنامج .

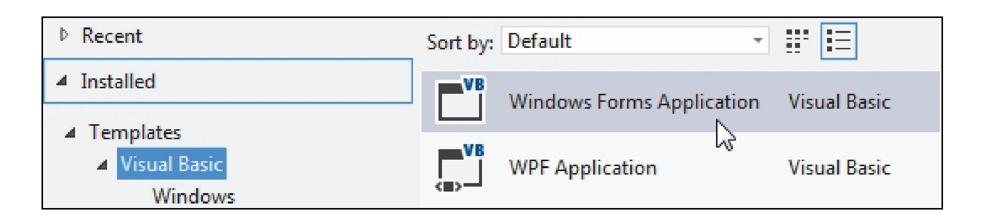
- وملفات الميديول Module .. وهى ملفات بها أكواد لأداء وظائف محددة تأخذ الإمتداد BAS عند تخزينها فى ذاكرة الكمبيوتر .

■ إنشاء مشروع جديد New Project :

1 – مـن الـقـائمة File إختار الأمر New Project .. أو بالضغط على مفتاحى **Shift + Ctrl + N**.



2 – تظهر النافذة New Project .. إختار منها Visual Basic ، ثم إضغط على الإختيار Windows Form Application .



3 – أكتب أسم المشروع الجديد فى المستطيل Name .. ثم حدد مكان حفظ المشروع فى المستطيل Location .. كما بالشكل التالى :

Name:	WindowsApplication7
Location:	c:\users\loly\documents\visual studio 2012\Projects
Solution:	Create new solution
Solution name:	WindowsApplication7

4 – إضغط **OK** فتظهر نافذة المشروع الجديد بأسم Form 1 .

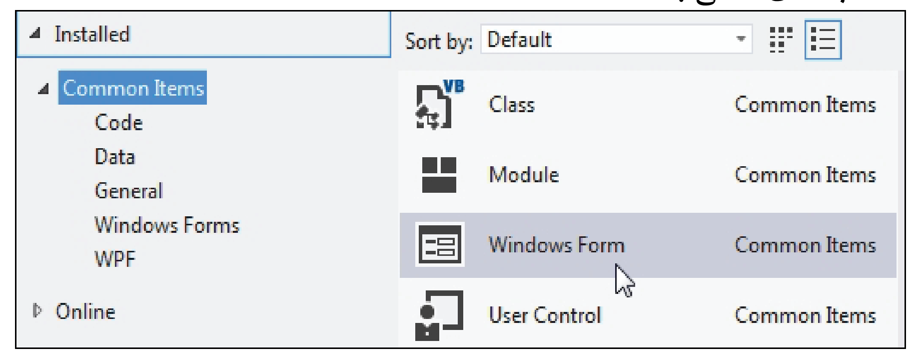
■ إضافة نموذج جديد للمشروع الحالى Add Form :

1 – من القائمة Project .. إختار Add Window Form ..
أو إختيار Add New Item ..
أو الضغط على مفاتيح **Shift + Ctrl + A** .

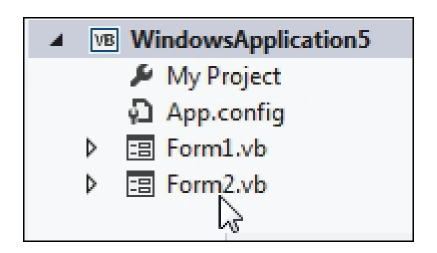
PROJECT

Add Windows Form...	
Add User Control...	
Add Module...	
Add Class...	Shift+Alt+C
Add New Data Source...	
Add New Item...	Ctrl+Shift+A
Add Existing Item...	Shift+Alt+A

2 – تظهر النافذة Add Form .. إختار منها Window Form .. كما بالشكل التالى :

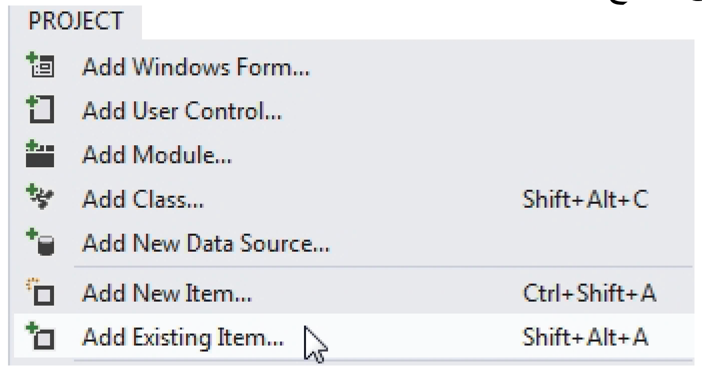


3 – إضغط مـفتاح Add فيظهر النموذج الجديد بأسم Form 2 فى النافذة Windows Application يمين نافذة البرنـامج .. كما بالشكل التالى :

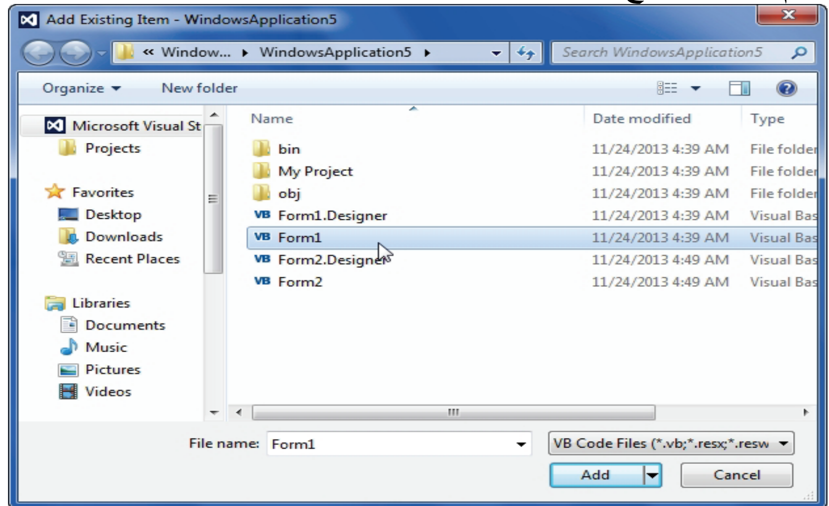


■ إضافة ملف مخزن من قبل للمشروع الحالى :

1 – من القائمة Project إختار الأمر Add Existing Item .. أو بالضغط على مفاتيح **Shift + Alt + A** ..

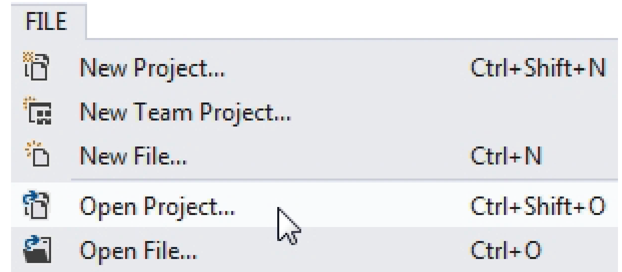


2 – تظهر النافذة Add Existing Item .. إختار الملف الذى تريد إضافتة ثم إضغط مفتاح **Add** ..

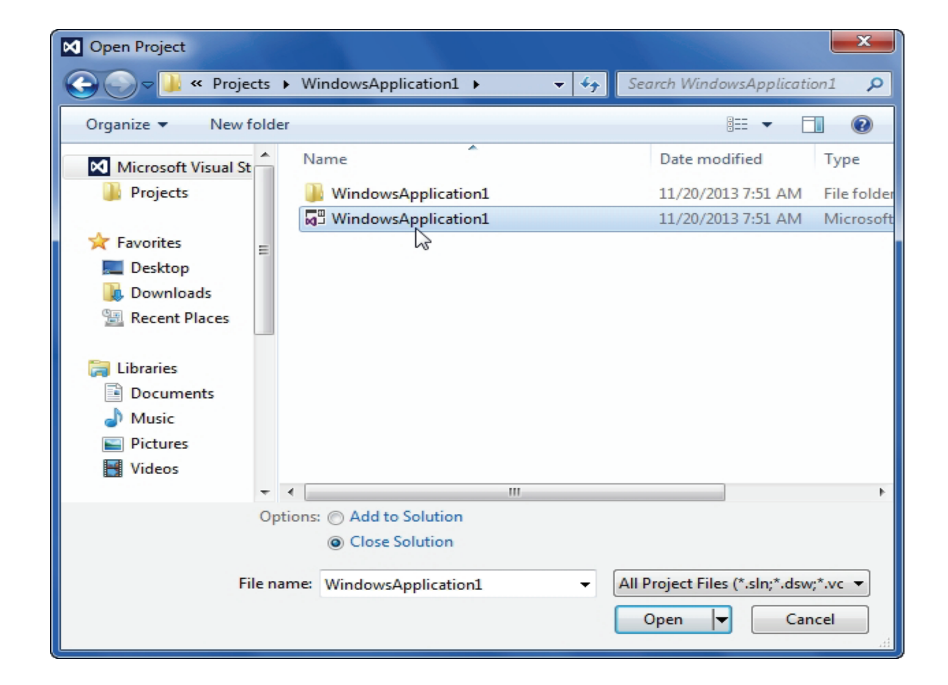


■ فتح مشروع موجود (سابق) لإستكمال العمل عليه أو تعديله :

1 – من القائمة File إختار الأمر Open Project .. أو بالضغط على مفاتيح **Shift + Ctrl + O** .



2 – تظهر النافذة Open Project .. ثم إختار منها أسم المشروع المطلوب فتحه ..



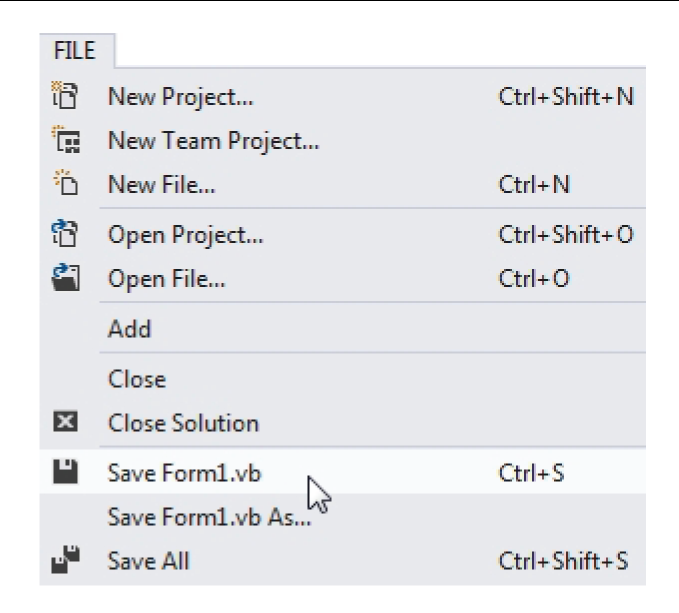
3 – إضغط مفتاح **Open** فيتم إضافة المشروع لبرنامج الـ Visual Basic .

■ حفظ ملفات المشروع الداخلية :

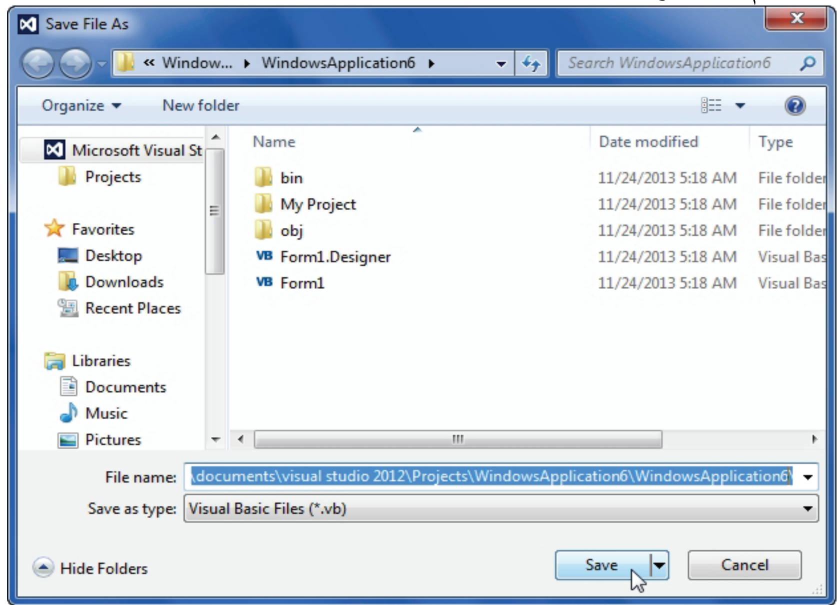
هنا يجب أن نفرق بين إختيارين لعملية الحفظ :

- **الأول :** حفظ ملفات المشروع لأول مرة .
- **الثانى :** حفظ ملفات مشروع مخزن من قبل .. ونريد حفظ التعديلات عليه .

1 – من القائمة File إختار Save Form 1 لحفظ ملفات المشروع الحالى بنفس الأسم ، أو Save Form As لحفظ ملفات المشروع لأول مرة ، أو المشروع الحالى بأسم جديد .



2 – تظهر نافذة مكان حفظ ملفات المشروع على جهاز الكمبيوتر .. إختار الأسم الذى تريده ..



3 – إضغط **Save** لتتم عملية الحفظ .

■ تصميم واجهة التطبيق Interface :

هى أولى خطوات عمل أى برنامج داخل لغة الـ Visual Basic .. وواجهة التطبيق هى الجزء المرئى من البرنامج الذى يتعامل من خلاله المستخدم مع البرنامج ، والمقصود هنا بواجهة التطبيق هى مجموعة من النماذج Forms ، وأدوات التحكم Controls ، والعناصر ، والقوائم التى يتكون منها البرنامج والتى لها مجموعة من :

- الخصائص Properties .
- الأساليب Methods .
- الأحداث Events .

وكل ما سبق يتحكم فى شكل وتسلسل أحداث البرنامج .

■ قبل البدء فى تصميم واجهات البرنامج لابد من معرفة الآتى :

1 – كم عدد النماذج Forms التى سيحتاجها البرنامج ؟
2 – ما هى الأوامر التى ستحتويها القوائم .. إن وجدت ؟
3 – هل ستستخدم أشرطة للأدوات .. لكى تؤدى بعض مهام القوائم أم لا ؟
4 – هل ستستخدم صناديق الحوار Dialog Box ؟ وفى أى مهمة ستستخدم ؟
5 – ما هو حجم المساعدة التى نريد تقديمها إلى المستخدم ؟
6 – ما هى نوعية المستخدم ذاته ؟

- هل هو مبتدئ ؟ يحتاج واجهة بسيطة وسهلة .
- أم محترف ؟ يحتاج لمرونة أكثر .

- **إذن :** واجهة التطبيق Interface تتكون من :
  - النماذج Forms .
  - القوائم Menus .
  - الأدوات والعناصر Controls .
  - صندوق الحوار Dialog Box .

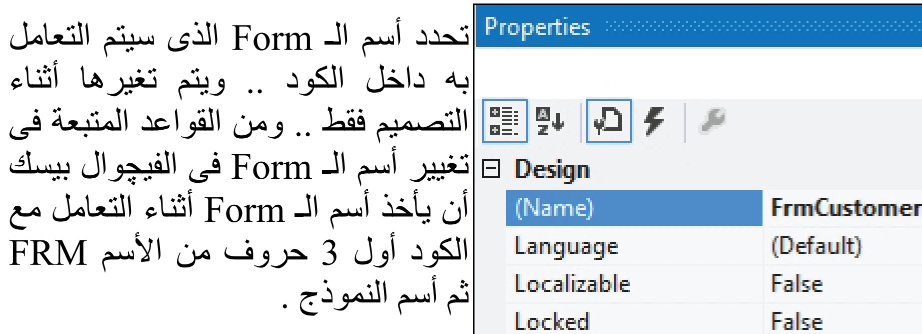
■ التعرف على النماذج Forms :

وهى أولى خطوات تصميم البرنامج ولها الشكل الواضح كما فى الشاشة التالية :



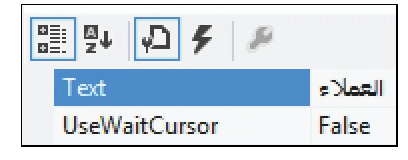
■ التعرف على لوحة الخصائص Properties :

• الخاصية Name :



تحدد أسم الـ Form الذى سيتم التعامل به داخل الكود .. ويتم تغيرها أثناء التصميم فقط .. ومن القواعد المتبعة فى تغيير أسم الـ Form فى الفيجوال بيسك أن يأخذ أسم الـ Form أثناء التعامل مع الكود أول 3 حروف من الأسم FRM ثم أسم النموذج .

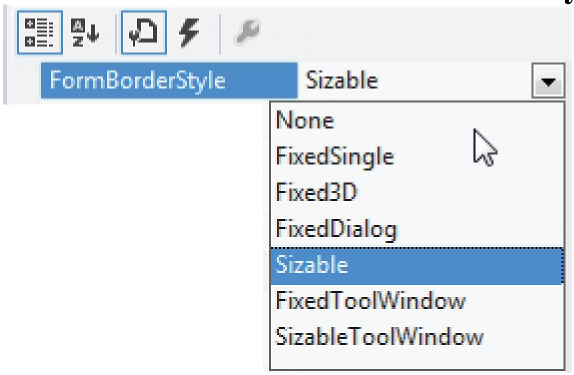
• Text :



تحدد النص الذى سيظهر كعنوان للنموذج Form فى شريط العنوان Tittle Bar .. وتظهر كما بالشكل التالى :



• الخاصية Form Border Style :



تستخدم لتحديد الإطار المحيط بالنموذج Form .. وبــها مجموعة من القيم من أهمها :

• **None :**

إخفاء الإطار الذى يحتوى على شريط العنوان Tittle Bar ومفاتيح التكبير والتصغير .

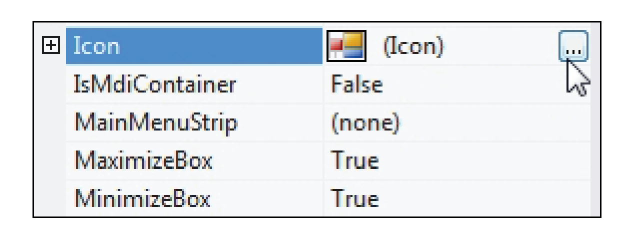
• **Fixed Single :**

يظهر شريط العنوان ومفتاح الإغلاق فقط .. ولا يمكن تغيير حجم الـ Form أثناء التشغيل .

• **Sizable :**

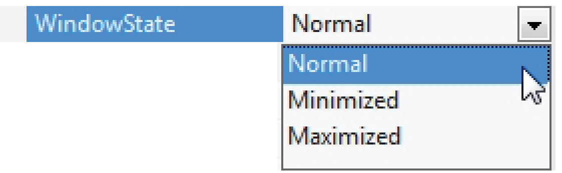
هى القيمة الإفتراضية للـ Form .. وتظهر النافذة فيه بالشكل الطبيعى .. أى ظهور مفاتيح تكبير وتصغير النموذج .

• **الخاصية Icon :**



و هى تحدد شكل الأيقونة التى ستظهر على الـ Form وبالضغط على الأداة الموجود يمين الخاصية Icon ستظهر نافذة .. إختار منها مكان صورة الأيقونة من على جهاز الكبيوتر بإمتداد ICO ، أو CUR .

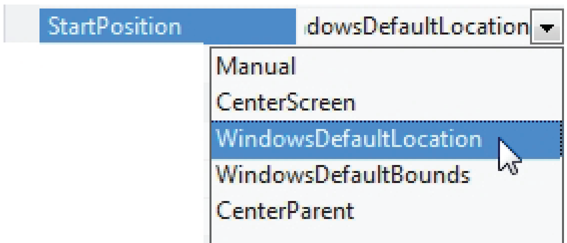
• الخاصية **Window State :**



تستخدم لتحديد حالة الـ Form عند بدء التشغيل .. ومنها :

• **Normal** .. الحجم الطبيعى .
• **Maximized** .. حجم يملئ الشاشة .
• **Minimized** .. عمل تصغير لتبقى على شريط المهام Task Bar بمجرد تشغيل البرنامج .

• الخاصية **Start Position :**



تحدد مكان الـ Form على الشاشة عند بدء تشغيل البرنامج .. ولها قيم منها :

• **Manual** .. تظهر الـ Form فى المكان المحدد بالقيم Top ، و Left .
• **Center Screen** .. وضع الـ Form فى منتصف الشاشة .

• **Windows Default** .. تقوم بوضع الـ Form فى مكان إفتراضى يقوم الـ Windows بتحديده .

• **الخاصية Size :**

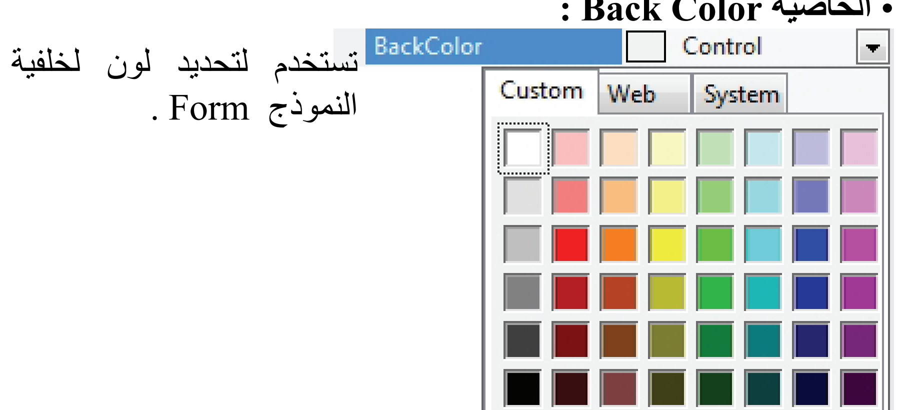
⊟ Size	401, 334
Width	401
Height	334

تحدد مقاس الـ Form على الشاشة من خلال القيمتين :

• **Width** .. لتحديد مقاس العرض .

• **Height** .. لتحديد مقاس الطول .

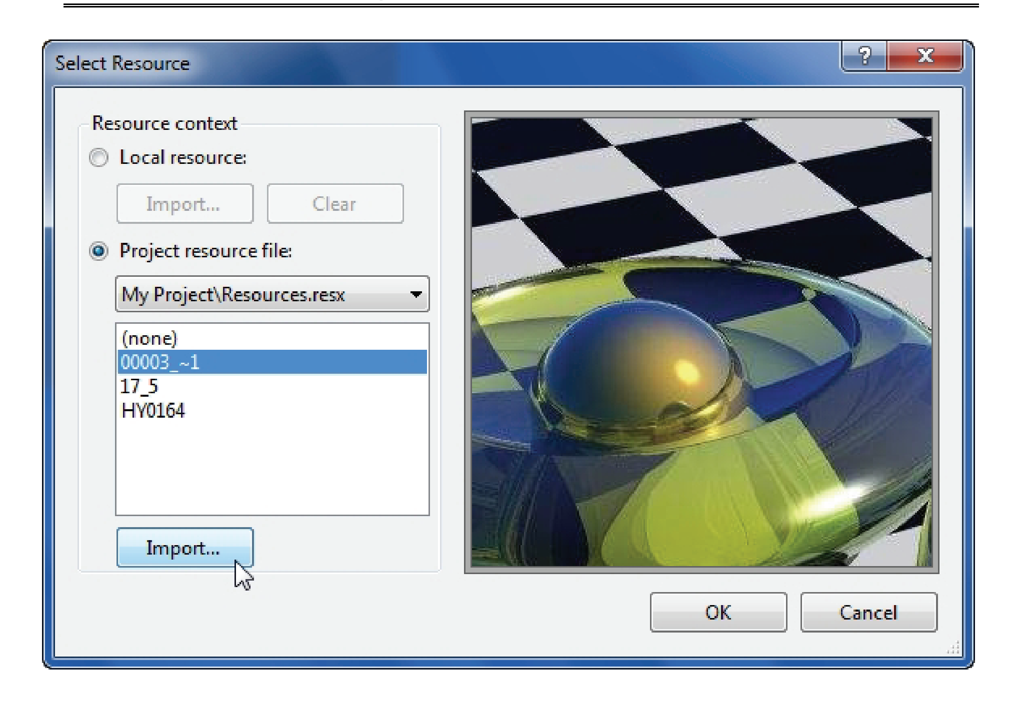
• **الخاصية Back Color :**



تستخدم لتحديد لون لخلفية النموذج Form .

• **الخاصية Back Ground Image :**

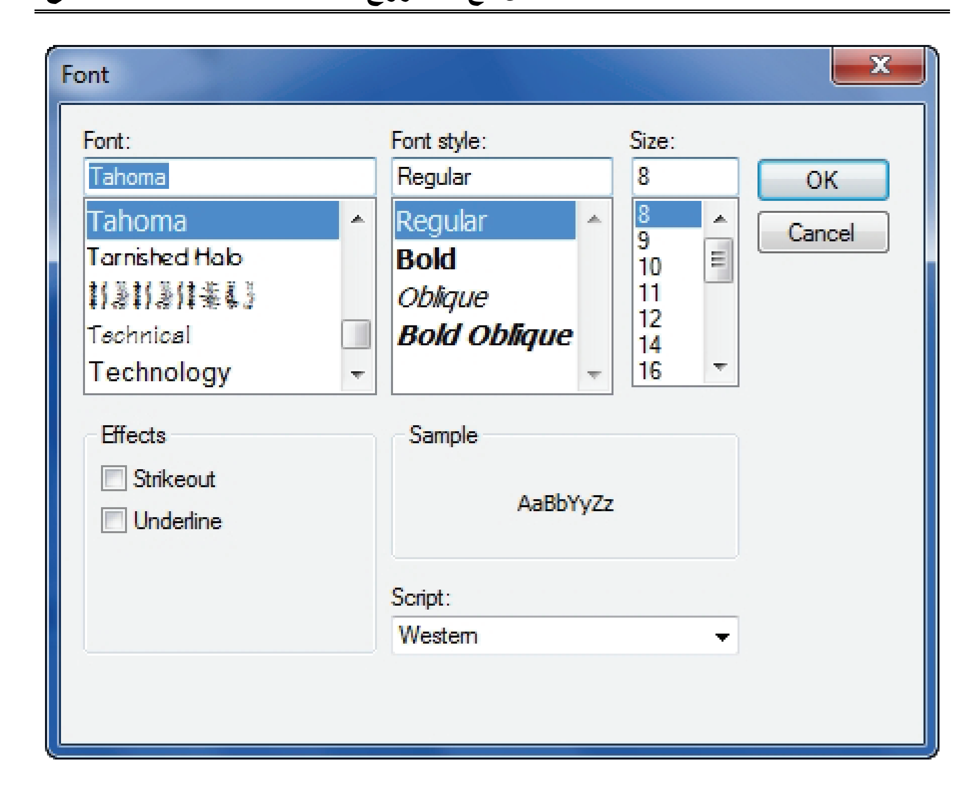
تستخدم لتحديد أى صورة كخلفية للنموذج Form .. وبالضغط على الأداة ... الموجودة بجانب الخاصية Back Ground Image تظهر النافذة التالية :



ومنها إضغط على المفتاح **Import** .. ثم حدد الصورة المطلوب تعيينها كخلفية للنموذج .. ثم إضغط **OK** .

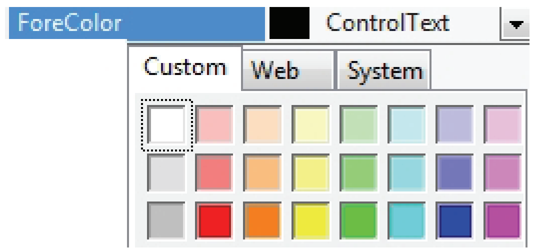
- **الخاصية Font :**

تستخدم لتحديد نوع ، وحجم ، وشكل الخط المستخدم فى النموذج Form وبالضغط على الأداة [...] الموجودة بجانب الخاصية Font تظهر النافذة التالية :



ومنها حدد أسم الخط من القائمة Font ، وشكل الخط من القائمة Font Style وحجم الخط من القائمة Size .. ثم إضغط **OK** .

- **الخاصية Fore Color :**



تستخدم لتحديد لون الخط الذى سيظهر على النموذج Form أثناء تشغيل البرنامج .



سوف يتم تناول باقى الخصائص بالتفصيل فى التطبيقات والأمثلة فى الفصول القادمة .

■ الأساليب Methods :

عبارة عن وسائل لأداء وظائف فى البرنامج .. مثل :

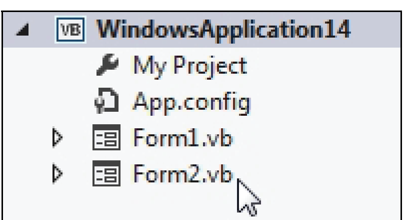
- **Show** .. لإظهار النافذة .
- **Load** .. لتحميل الـ Form داخل الذاكرة .
- **Hide** .. لإخفاء النافذة .
- **Unload** .. لإزالة الـ Form من الذاكرة .

ولتوضيح الوظائف التى تقوم بتنفيذها الأساليب Methods داخل البرنامج من خلال المثال التالى :



1 – إفتح مشروع جديد .. وذلك من القائمة File ثم إختار New Project ، أو بالضغط على مفاتيح **Shift + Ctrl + N** .

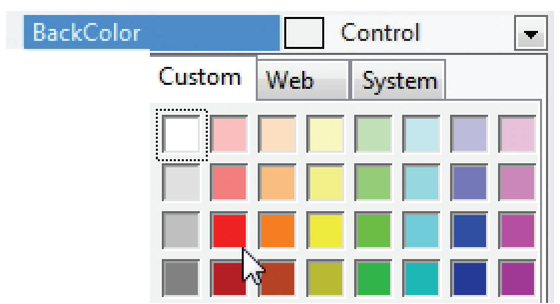
2 – قم بإضافة Form جديد للمشروع .. وذلك مـن القائمة Project ثم إخـتار Add New Item ، أو بالضغط على مفاتيح **Shift + Ctrl + A** .. فيظهر فى النافذة Windows Application الموجودة يمين البرنامج عدد وأسماء الـ Forms الموجودة فى المشروع .. كما بالشكل التالى :



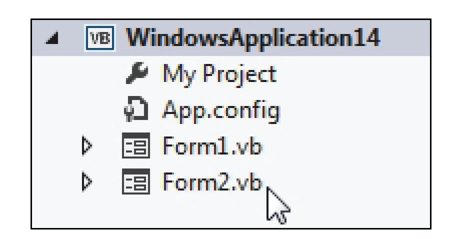


الأسم الإفتراضى لأول Form داخل المشروع هو Form 1 .. والنموذج التالى يأخذ الأسم Form 2 .. وهكذا حسب عدد الـ Forms المكون منها المشروع .. ويتم بعد ذلك تغيير أسم الـ Form بالأسم المناسب من الخاصية Text .

3 – غير لون خلفية الـ Form 1 .. وذلك بالضغط على الـ Form مرة واحده .. ثم من نافذة الخصائص إختار اللون الأحمر من الخاصية Back Color .

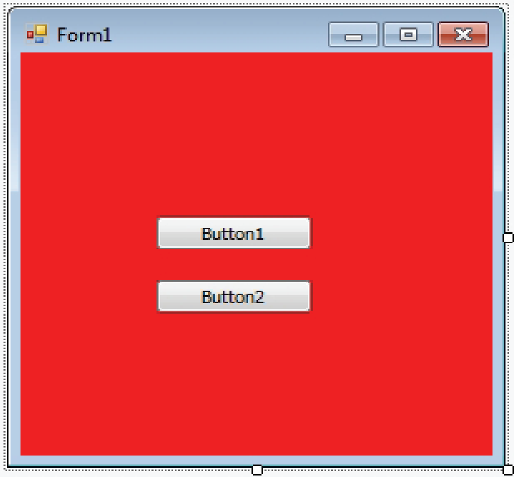


4 – حدد Form 2 من النافذة Windows Application ..



ثم غير لون الخلفية من نافذة الخصائص باللون الأزرق .. كما فى الخطوة السابقة .

5 – حدد Form 1 .. ثم قم بإضافة عد 2 مفتاح Button من صنـدوق الأدوات Tool Box على الـ Form 1 .. كما بالشكل التالى :



6 – غير أسماء المفتاحين السابقين .. من الخاصية Text ليصبح كل مفتاح على التوالى :

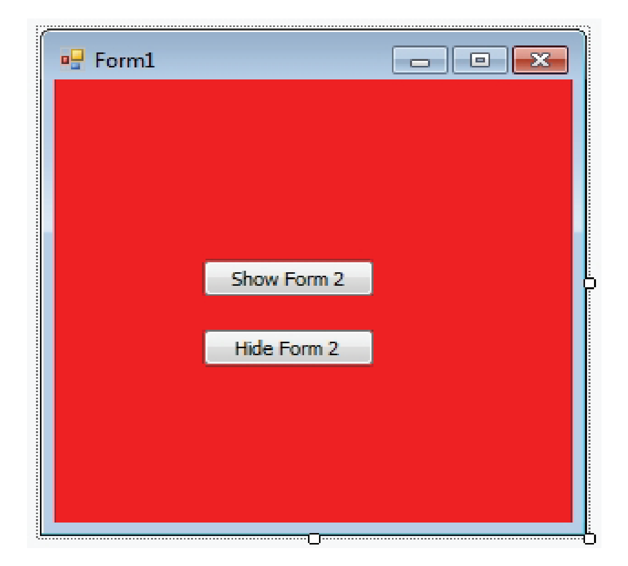
- **Show Form 2 :**

وتعنى إظهار النافذة Form 2 بمجرد الضغط على هذا المفتاح .

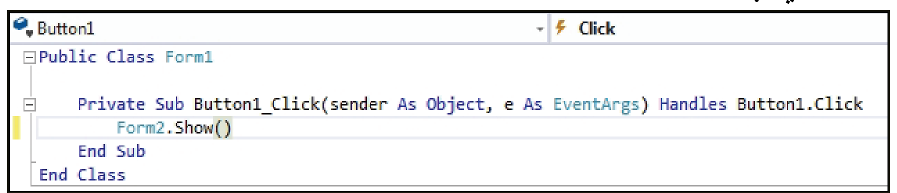
- **Hide Form 2 :**

وتعنى إخفاء الـ Form 2 بمجرد الضغط على هذا المفتاح .

وتظهر النافذة Form 1 بعد تغيير أسماء المفاتيح .. كما بالشكل التالى :



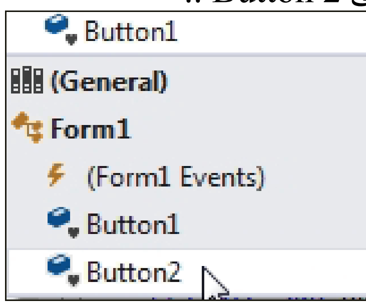
7 – إضغط على المفتاح الأول Show Form 2 مرتين .. فتظهر نافذة الكود التالية :



ثم أكتب بين الجملتين Private Sub ، وEnd Sub الكود التالى :

```
Form2.Show
```

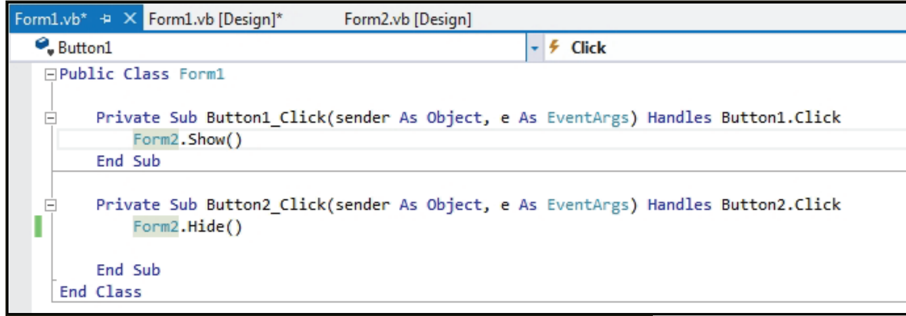
8 – من القائمة المنسدلة Object الموجودة أعلى يسار نافذة الكود .. إختار المفتاح الثانى Button 2 ..



ثم أكتب الكود التالى :

Form2.Hide

ويكون الشكل النهائى للجمل داخل نافذة الكود .. كما بالشكل التالى :



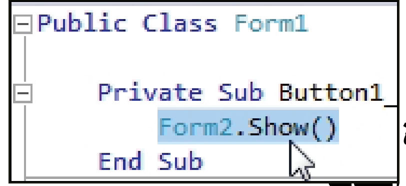
```
Public Class Form1

    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        Form2.Show()
    End Sub

    Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
        Form2.Hide()

    End Sub
End Class
```



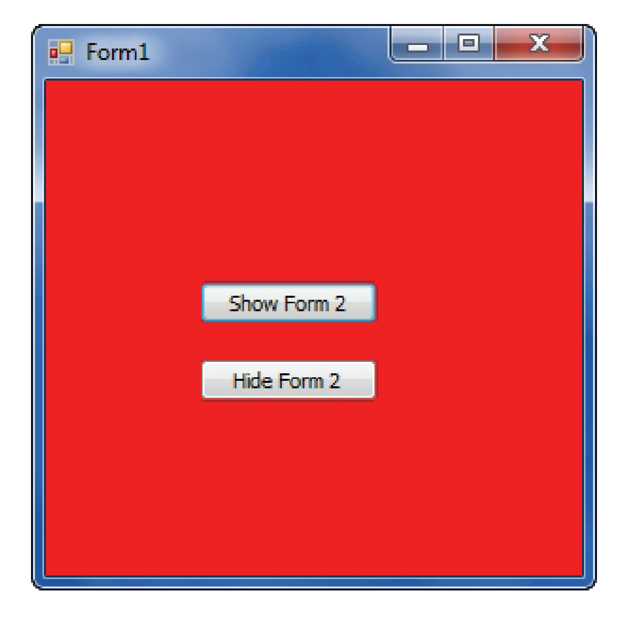


```
Public Class Form1

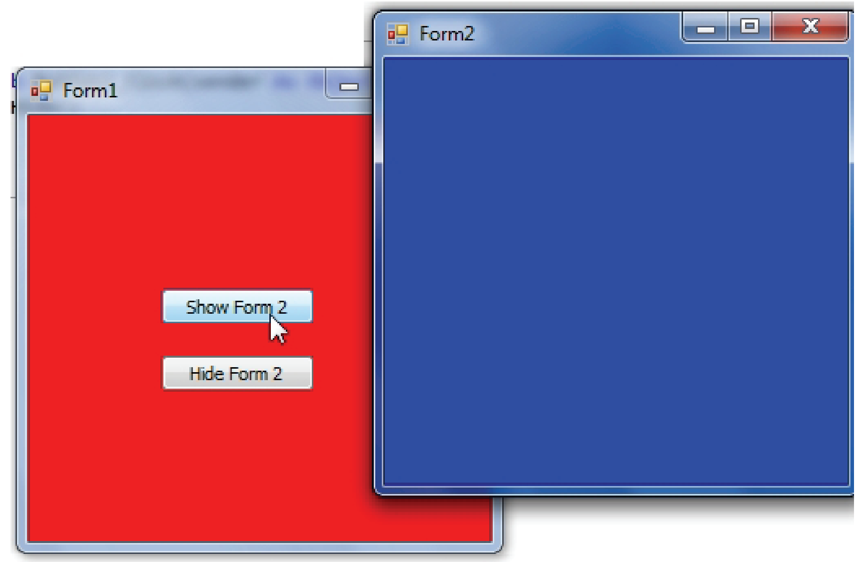
    Private Sub Button1
        Form2.Show()
    End Sub
```

- ما يتم كتابته فى نافذة الكود هى الجمل التى بين المستطيل .. وليس كتابة كل نصوص صفحة الكود البرمجي كما يعتقد البعض
- يمكن نسخ الأكواد السابقة بدلاً من إعادة الأكواد المرفق بإسطوانة الـ **CD** مع هذا الكتاب .

9 – شغل البرنامج وذلك بالضغط على المفتاح Start **Start** الموجود على شريط الأدوات لأعلى نافذة البرنامج .. أو بالضغط على مفتاح **F5** .. فيظهر البرنامج كما بالشكل التالى :



10 – إضغط على المفتاح Show Form 2 فتظهر النافذة Form 2 كما بالشكل التالى :



- إضغط على مفتاح Hide Form 2 فتختفى النافذة Form 2 .



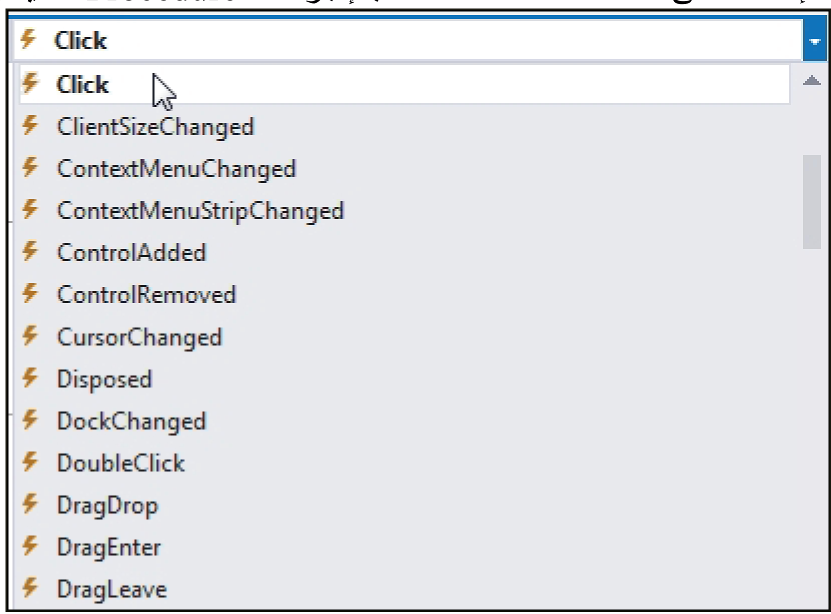
يتم إستخدام خطوات المثال السابق فى إنشاء وتنفيذ برنامج متكامل .. وذلك من خلال الضغط على المفاتيح المكون منها البرنامج لإظهار النماذج الداخلية للبرنامج .. بالإضافة إلى وظائف أخرى للمفاتيح .. سوف نتعرف عليها فى الفصول القادمة .

■ الأحداث Events :

هى أحداث مرتبطة بالـ Form .. ويمكن الوصول للأحداث Events بالخطوات التالية :

1 – إضغط مرتين على الـ Form لفتح نافذة الكود .

2 – إضغط على القائمة المنسدلة الخاصة بالإجراءات Procedure التالية :



وتحتوى القائمة Procedure على جميع الإجراءات والأحداث المتعلقة بالـ Form .. والتى من أهمها :

■ الحدث Load :

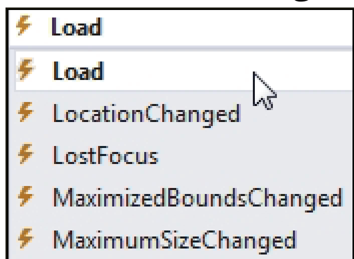
ويحدث نتيجة تحميل الـ Form .. ويستخدم هذا الحدث إذا أرادنا أداء وظيفة معينة تبدأ مع تحميل الـ Form مثل عمل تنشيط لبعض الـ Controls أو العناصر الموجودة على الـ Form بناءاً على قيمة بعض المتغيرات ، أو لكتابة بعض النصوص الإفتراضية داخل بعض الـ Text Box .



يعتبر الحدث Load .. هو حدث تلقائى للـ Form .. أى يظهر تلقائياً عند فتح نافذة الكود الخاصة بالـ Form داخل القائمة Procedure .

- ولتنفيذ إستخدام الحدث Load داخل البرنامج إتبع الخطوات التالية :

1 – إضغط مرتين على نافذة الـ Form .. فتظهر نافذة الكود .
2 – إختار الحدث Load من القائمة المنسدلة Procedure ..



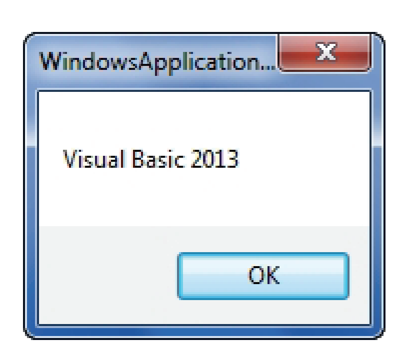
3 – أكتب الجملة التالية داخل نافذة الكود عند موضع مؤشر الـ Mouse داخل مكان الحدث ..

```
MsgBox "Visual Basic 2013"
```

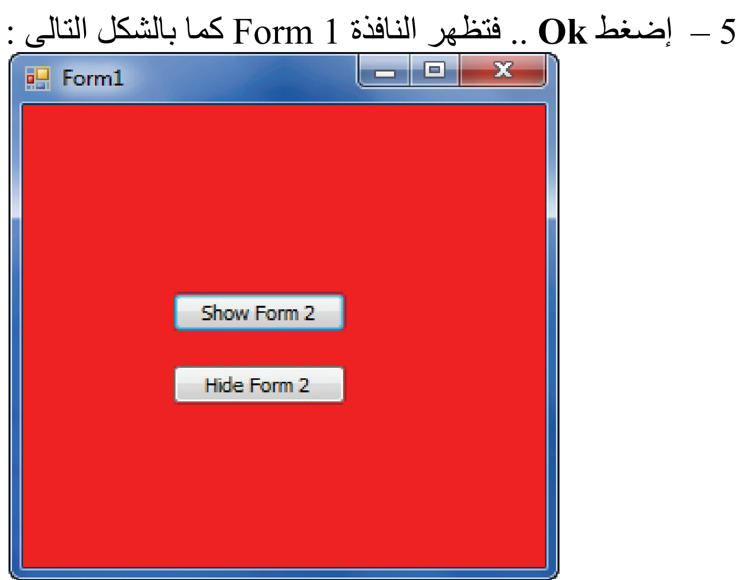
```
Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
    MsgBox("Visual Basic 2013")
End Sub
End Class
```

وتعنى الجملة السابقة .. ظهور رسالة تحتوى على عنوان الـ Form .. قبل ظهور الـ Form نفسه .

4 – شغل البرنامج بالضغط على مفتاح **F5** فتظهر رسالة تحتوى على كلمة **Visual Basic 2013** قبل بداية ظهور الـ Form .. كما بالشكل التالى :



5 – إضغط **Ok** .. فتظهر النافذة Form 1 كما بالشكل التالى :

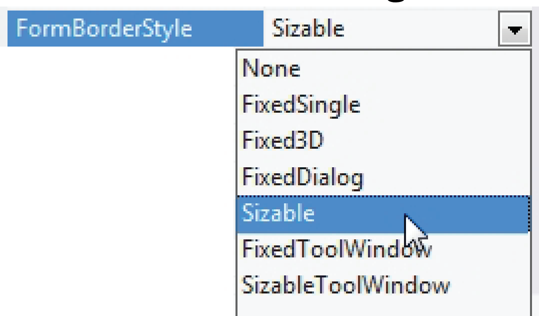




الحدث Load ينفذ عند بدء تحميل النافذة وليس عند ظهورها .

■ **الحدث Resize :**

يقوم هذا الحدث بتغيير حجم الـ Form .. بشرط أن تكون الخاصية Border Style Form هى Sizble ..



ويمكن الإستفادة بهذا الحدث لعمل إعادة لأحجام الـ Controls الموجودة على الـ Form عند تغيير حجمها .. بحيث تكون أحجام الـ Controls نسبية بين إرتفاع ، وعرض الـ Form على الشاشة .

■ **الحدث Click :**

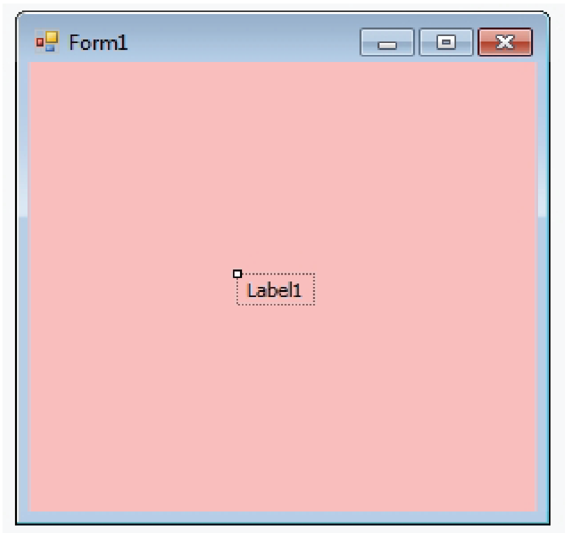
يستخدم لتنفيذ الحدث فى حالة الضغط مرة واحدة فقط على العنصر .. مثل الضغط على مفتاح التحكم لتنفيذ الأمر .. فمثلاً عند فتح نافذة الكود الخاصة بالمفتاح Button1 يظهر تلقائياً الحدث Click لهذا العنصر .

■ **الحدث Double Click :**

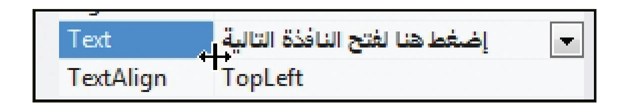
يستخدم لتنفيذ الحدث فى حالة الضغط مرتين على العنصر .. ولتنفيذ ذلك إتبع الخطوات التالية :

1- أضف عدد 2 Form للمشروع الحالى .. ثم غير لون الخلفية لكل Formباللون المناسب.

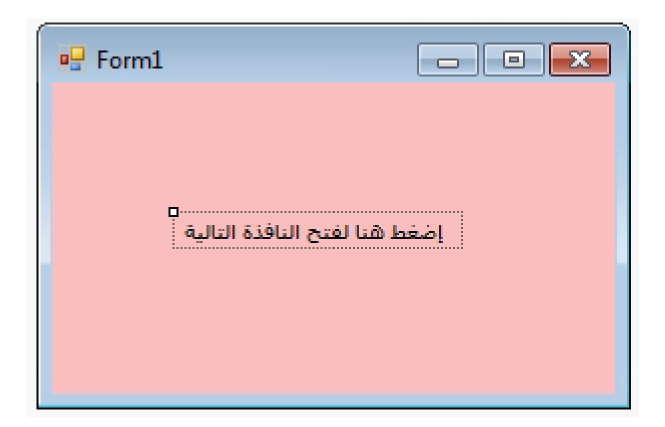
2 – أضف الأداه Label الخاص بكتابة عنوان للنموذج وهى Label 1 من صندوق الأدوات Tool Box .. فتظهر كما بالشكل التالى :



3 – أكتب النص المناسب للعنوان فى الخاصية Text .. كما بالشكل التالى :

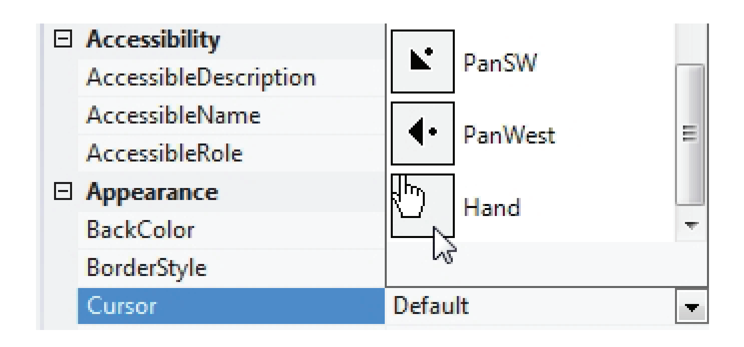


فيظهر النص كما بالشكل التالى :



4 – كرر الخطوة السابقة للنموذج الثانى Form 2 .

5 – غير شكل مؤشر الـ Mouse من الخاصية Cursor ليكون شكل علامة اليد Hand .. كما بالشكل التالى :

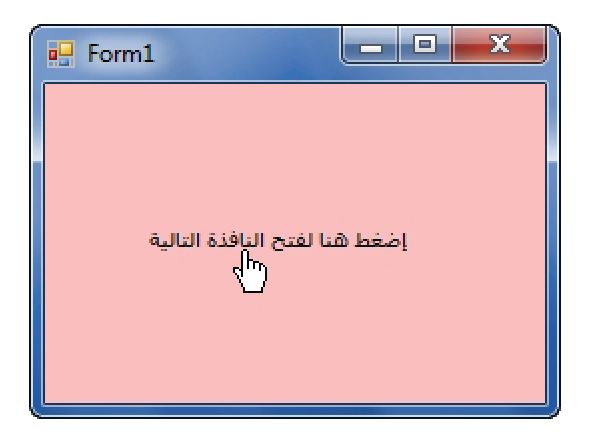


5 – إضغط مرتين على النص المكتوب فى النافذة Form 2 فتظهر نافذة كتابة الكود .

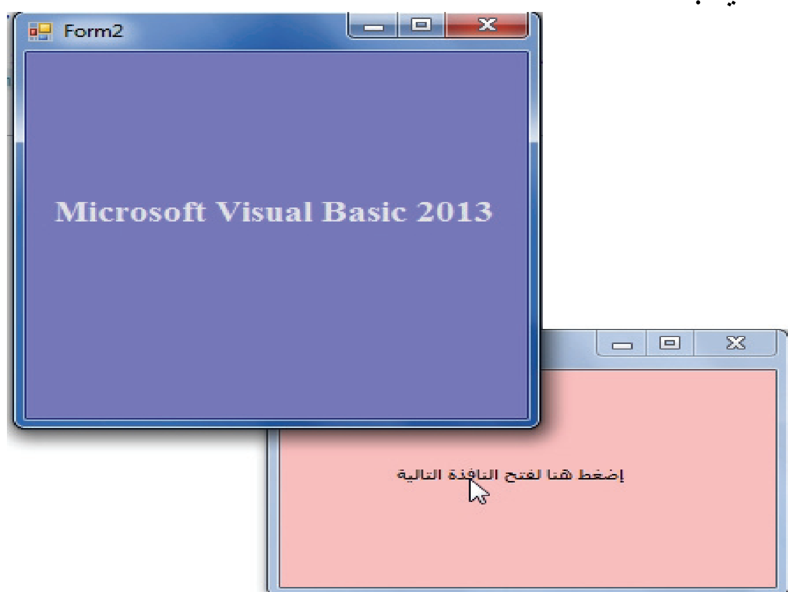
6 – إختار من القائمة المنسدلة Procedure الحدث Double Click ثم إكتب الكود التالى :

```
Form2.Show
```

7 – شغل البرنامج بالضغط على مفتاح **F5** .. فتظهر النافذة التالية :



8 – إضغط مرتين على الـنص المكتوب فى النافذة الـسابقة .. فتظهر النافذة التالية :





يستخدم الحدث Click ، والحدث Double Click لعمل روابط بين الـ Forms داخل البرنامج بدون إستخدام مفاتيح التحكم .

الفصل الخامس
التعامل مع صندوق الأدوات
Tool Box

نتعلم فى هذا الفصل ..



- إظهار وإخفاء صندوق الأدوات .
- عرض نافذة صندوق الأدوات Tool Box بأكثر من طريقة .
- إضافة العناصر Control إلى النموذج Form .
- تغيير الخصائص Properties الخاصة بالأدوات .
- أمثلة وتطبيقات على الأدوات المختلفة .

الفصل الخامس

التعامل مع صندوق الأدوات

Tool Box

يعتبر صندوق الأدوات Tool Box من الأدوات أو الـ Controls الهامة فى تصميم واجهة التطبيق لأى برنامج .. مثل مفاتيح التحكم ، وحقول إدخال البيانات ، وعناصر الإختيارات ، والقوائم المنسدلة .. وغيرها من الأدوات المكونة للبرنامج .

كما يحتوى كل عنصر ، أو Control على مجموعة من الخصائص Properties ومجموعة من الأساليب Methods ، ومجموعة من الأحداث Events .. كما تعلمنا ذلك فى الفصول السابقة .. وسوف نتناول ذلك بالتفصيل فى الصفحات القادمة .

■ إظهار وإخفاء صندوق الأدوات Tool Box :

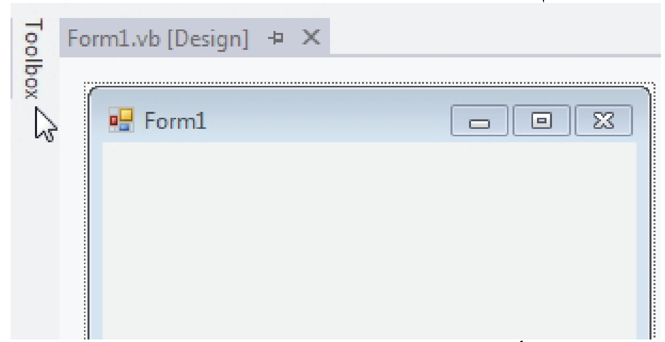
1 – من القائمة View إختار الأمر Toolbox .. أو بالضغط على مفاتيح **Ctrl + W,X** معاً من لوحة المفاتيح ..

VIEW

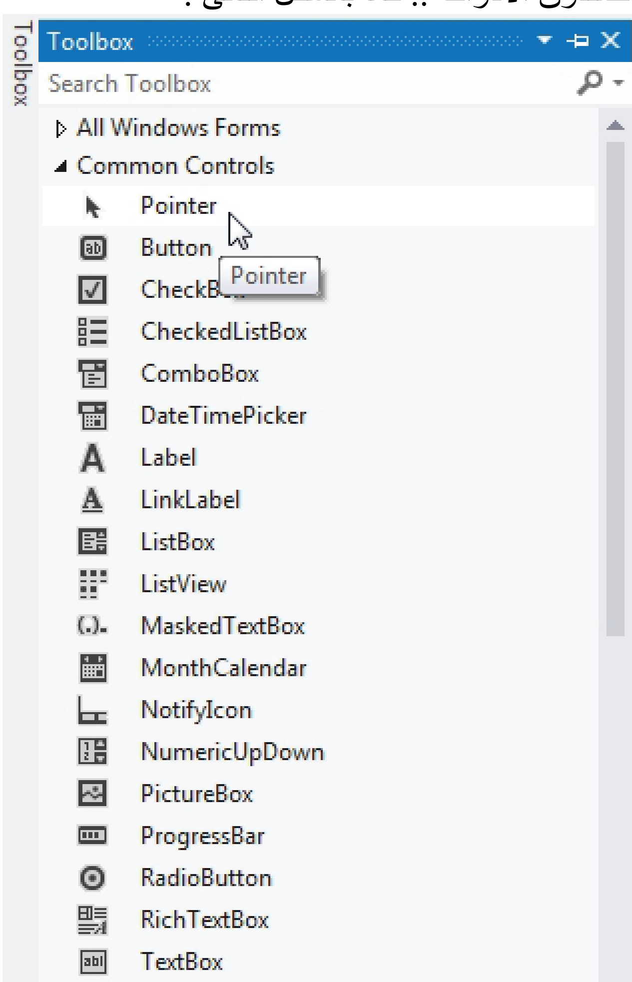
Code	Ctrl+Alt+0
Designer	Shift+F7
Start Page	
Solution Explorer	Ctrl+W, S
Team Explorer	Ctrl+\, Ctrl+M
Call Hierarchy	Ctrl+Alt+K
Object Browser	Ctrl+W, J
Class View	Ctrl+W, C
Error List	Ctrl+W, E
Output	Alt+2
Task List	Ctrl+W, T
Bookmark Window	Ctrl+W, B
Toolbox	Ctrl+W, X

2 – تظهر نافذة صندوق الأدوات Tool Box يسار نافذة البرنامج .

3 – إضغط على أسم النافذة Tool Box ..

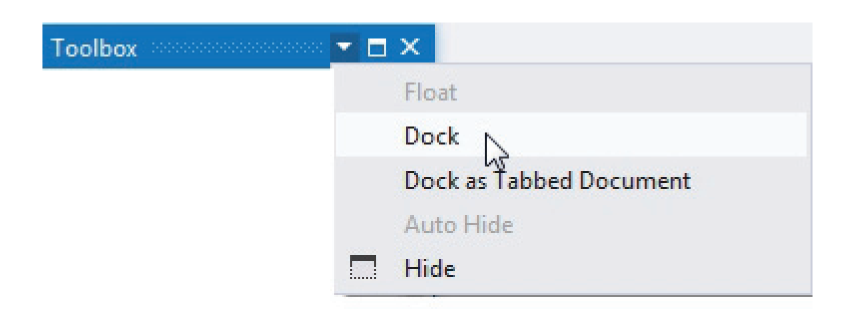


فتظهر نافذة صندوق الأدوات .. كما بالشكل التالى :



■ عرض نافذة صندوق الأدوات Tool Box بأكثر من طريقة :

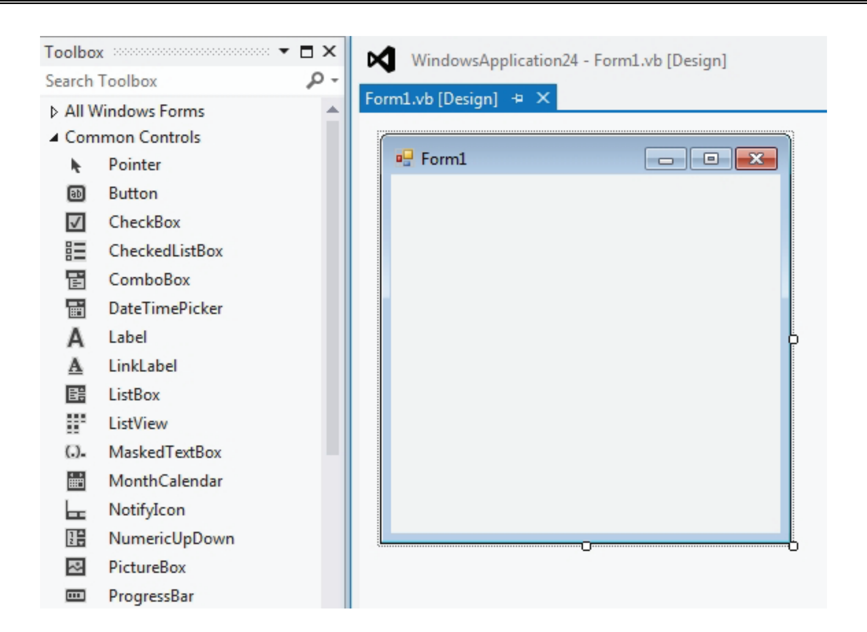
1 – إضغط على المثلث الصغير الموجود على أسم نافذة صندوق الأدوات Tool Box .. فتظهر قائمة مختصرة .. كما بالشكل التالى :



- ومنها :

- **Float** .. عرض وتحريك نافذة صندوق الأدوات فى أى مكان مناسب .
- **Dock** .. وضع النافذة بشكل منفصل بجانب الـ Form ، وذلك لإختيار الأدوات منها ووضعها داخل النموذج .
- **Hide** .. إخفاء النافذة Tool Box من شاشة البرنامج .
- **Auto Hide** .. إخفاء تلقائى للنافذة بعد إختيار الأداة منها .

2 – إختار الوضع المناسب لنافذة صندوق الأدوات على الشاشة .. فتظهر نافذة الـ Form ، ونافذة الـ Tool Box .. كما بالشكل التالى :

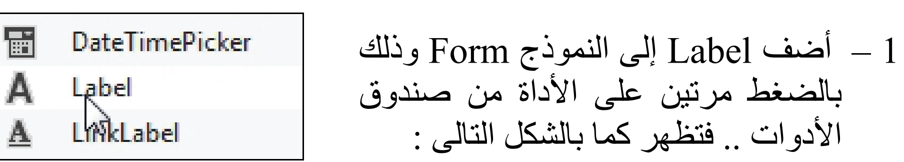


■ إضافة العناصر Controls إلى النموذج Form :

يؤثر الشكل الجيد للنموذج على المستخدم إثناء التعامل مع البرنامج .. ويتم ذلك من خلال تنسيق العناصر على النموذج ووضعها بالشكل المناسب الذى يسهل على المستخدم التعامل معه ..

■ الأداة Label A :

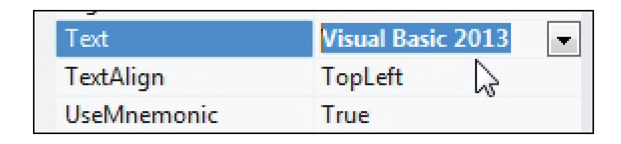
يستخدم الـ Label لوضع عنوان داخل النموذج .. كما أنه لايمكن التحكم فى هذا العنوان مباشرة من قبل المستخدم أو تغييره .. بل هو مجرد عنوان فقط ، ويحتوى العنوان Label على الخاصية Text التى من خلالها يتم كتابة النص الذى سيظهر على الـ Form .. ولتنفيذ ذلك .. إتبع الخطوات التالية :



1 – أضف Label إلى النموذج Form وذلك بالضغط مرتين على الأداة من صندوق الأدوات .. فتظهر كما بالشكل التالى :

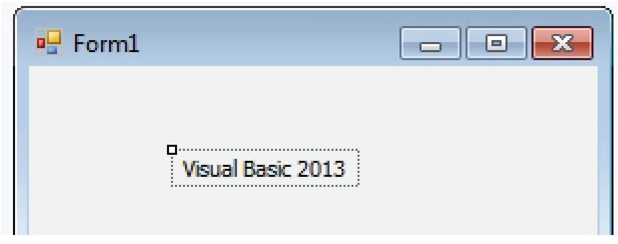


2 – أكتب أسم للعنوان فى الخاصية Text.. وليكن كلمة2013 Visual Basic

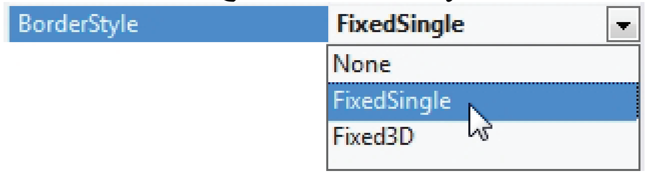


Text	Visual Basic 2013
TextAlign	TopLeft
UseMnemonic	True

فيظهر العنوان كما بالشكل التالى :

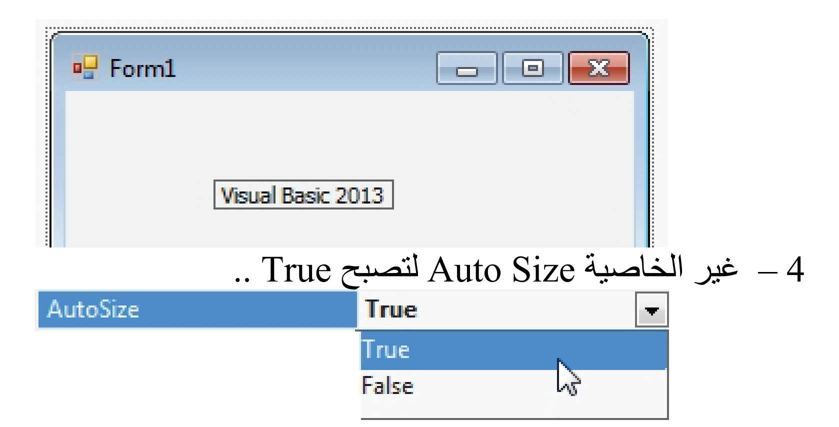


3 – غير قيمة الخاصية Border Style لوضع إطار حول الـ Label ..



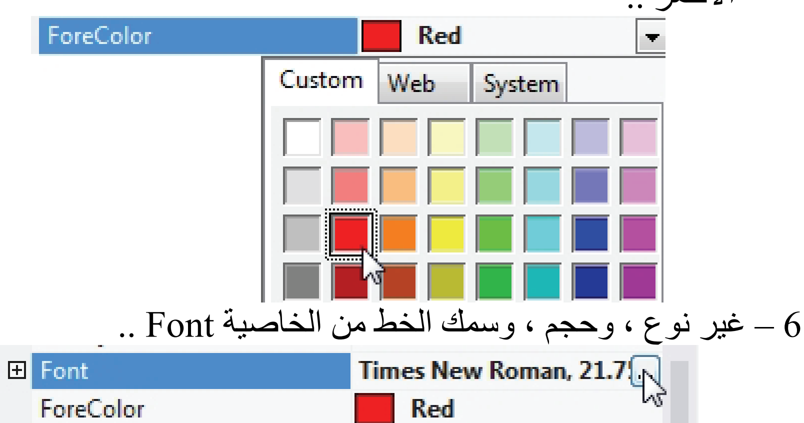
- **None** .. عدم وضع إطار .
- **Fixed Single** .. وضع إطار .
- **Fixed 3D** .. وضع إطار ثلاثى الأبعاد .

- إختار **Fixed 3D** .. فيظهر الإطار حول العنوان كما بالشكل التالى :



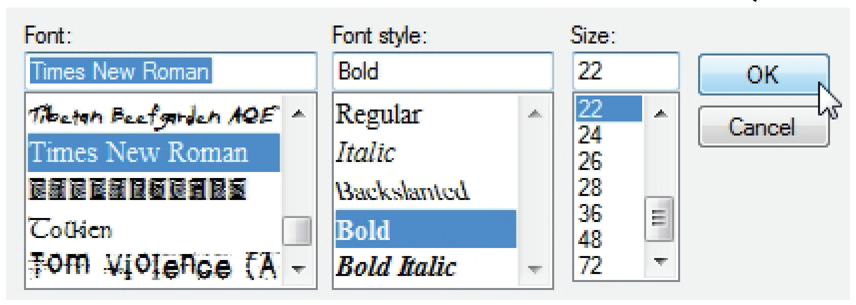
4 – غير الخاصية Auto Size لتصبح True ..

5 – إختار اللون المناسب للنص من الخاصية Fore Color .. وليكن اللون الأحمر ..



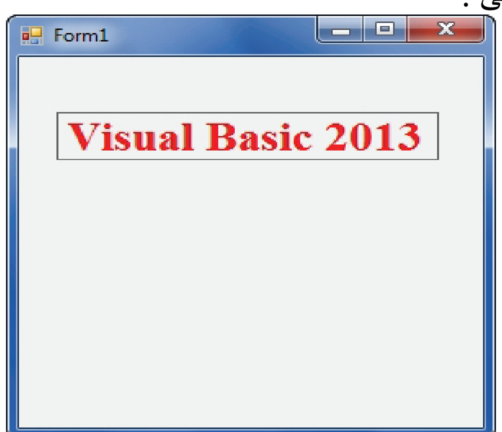
6 – غير نوع ، وحجم ، وسمك الخط من الخاصية Font ..

وبالضغط على المربع [...] الصغير الموجود بجانب الخاصية Font تظهر النافذة التالية :



إختار منها حجم الخط من القائمة Size ، شكل الخط من القائمة Font Style ، ونوع الخط من القائمة Font .. ثم إضغط **OK** .

7 – شــغل الـبرنـامج .. بالضغط على مفتاح **F5** .. فيظهر النموذج كـما بالشكل التالى :



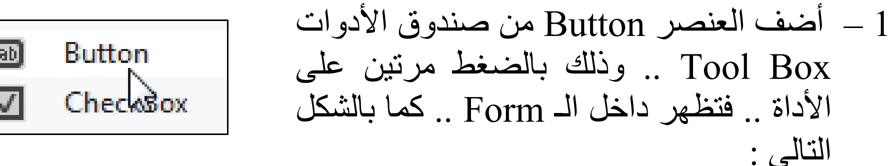


لا يوجد أى أحداث Events ، أو أساليب Methods خاصة بالعنوان Label فهو يحتوى على الـ Methods العامة مثل Move الخاصة بتحريك الـ Form على الشاشة أثناء تشغيل البرنامج .. ومثل Refresh الخاصة بإعادة كتابة النص داخله .

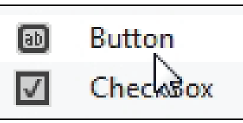
■ **الأداة Button :**

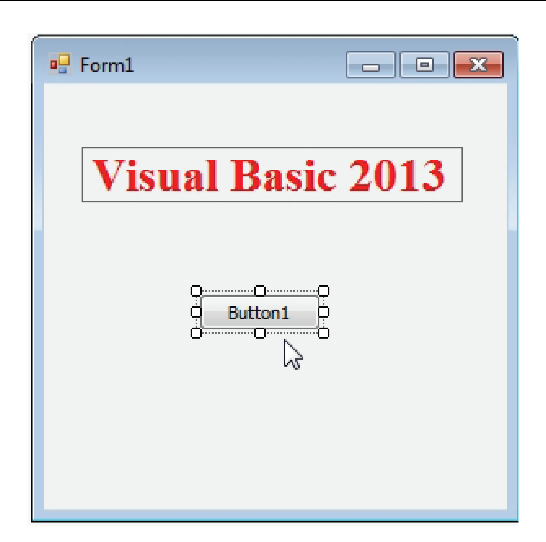
تستخدم لعمل مفاتيح التحكم داخل النموذج .. وذلك لتنفيذ أى عملية داخل البرنـامج .. مثل (الفتح / الإغلاق / الحفظ / الإلغاء) .. وأيضاً إجراء العمليات الحسابية .

ولتنفيذ إستخدام العنصر Button .. إتبع الخطوات التالية :

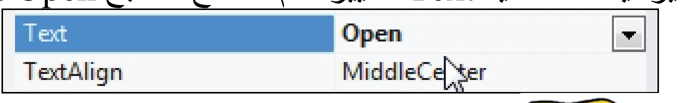


1 – أضف العنصر Button من صندوق الأدوات Tool Box .. وذلك بالضغط مرتين على الأداة .. فتظهر داخل الـ Form .. كما بالشكل التالى :





2 – غير قيمة الخاصية Text لتغيير أسم المفتاح لتصبح Open ..

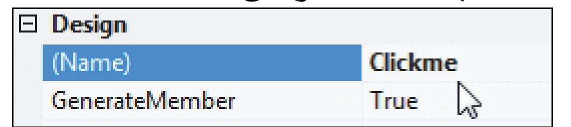


Text	Open
TextAlign	MiddleCenter



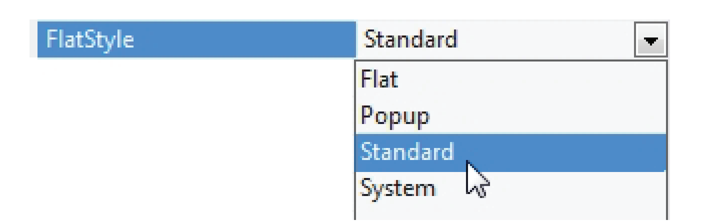
الخاصية Text .. هى التى تحدد أسم العنصر على النافذة Form ، أما الخاصية Name فهى تحدد أسم العنصر داخل الكود البرمجى .

3 – غير قيمة الخاصية Name .. ولتكن مثلاً كلمة Click Me ..



Design	
(Name)	Clickme
GenerateMember	True

4 – غير قيمة الخاصية Flat Style لتصبح Standard .. و هى شكل المفتاح على الشاشة ..

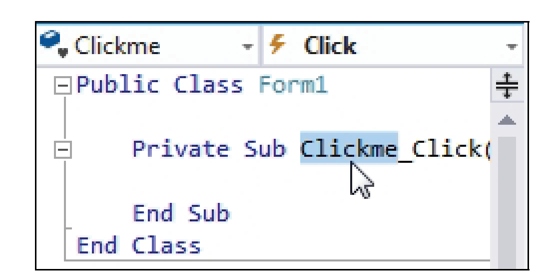




يمكن إختيار صورة لتظهر على المفتاح أثناء التشغيل وذلك بإختيار الصورة المناسبة من الخاصية Image .. لتظهر كما بالشكل التالى :



5 – إضغط مرتين على المفتاح Button .. فتظهر نافذة الكود .. فنجد أنه يظهر تلقائياً الحدث Click ، وأسم الإجراء Click Me كما بالشكل التالى :



6 – قم بكتابة الجملة التالية .. كرسالة تظهر عند الضغط على المفتاح أثناء تشغيل البرنامج :

MsgBox" مرحباً بكم في دار البراء لنشر وتوزيع الكتب والوسائط العلمية "

7 – شغل البرنامج بالضغط على مفتاح **F5** .. ثم إضغط على المفتاح Open فتظهر نافذة الرسالة .. كما بالشكل التالى :





توجد أيضاً وظيفتين أثناء تشغيل البرنامج للأداة Button تسمح بقراءة لوحة المفاتيح Keyboard وذلك للإستجابة للمفتاح Enter عند الضغط عليه يقوم بتنفيذ المفتاح النشط ، والمفتاح Esc عند الضغط عليه يقوم بإلغاء التنفيذ .

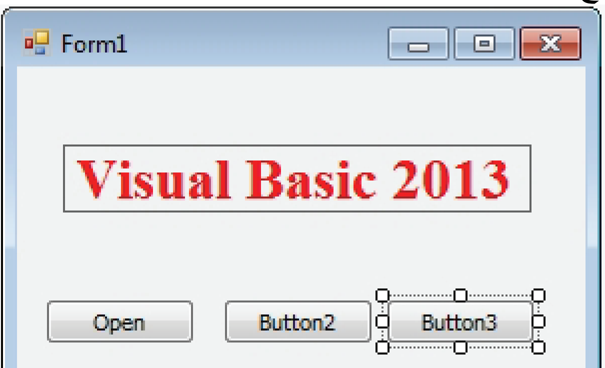
س / ماذا لو هناك أكثر من مفتاح فى نفس الوقت داخل النموذج الواحد ؟

هنا يتم إضافة أكثر من مفتاح حسب عدد المفاتيح المطلوبة .. ويتم كتابة كود لكل مفتاح حسب المهمة المطلوبة منه .. ولتنفيذ ذلك من خلال المثال التالى :

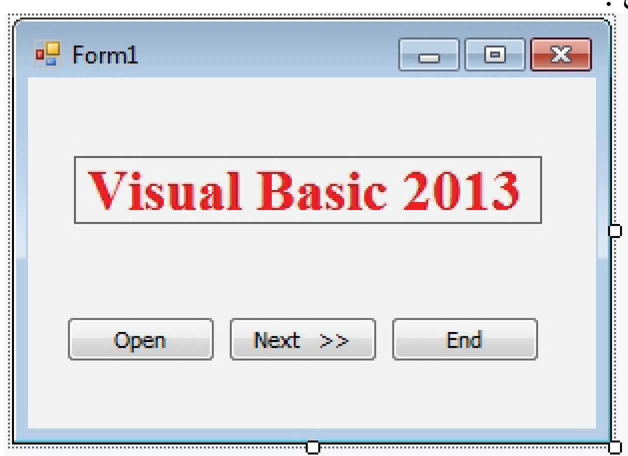


صمم برنامج يتم من خلاله تنفيذ مجموعة من المهام عند الضغط على المفاتيح الموجوده داخل النموذج .. ولتنفيذ ذلك .. إتبع الخطوات التالية :

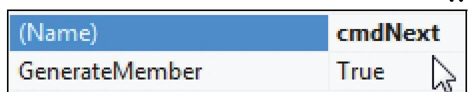
1 – أضف عدد 2 مفتاح للنموذج السابق .. ليصح أمامنا 3 مفاتيح داخل النموذج .. كما بالشكل التالى :



2 – غير الخاصية Text للمفتاح الثانى لتصبح **Next >>** ، وغير الخاصية Text للمفتاح الثالث لتصبح **End** .. فيظهر النموذج كما بالشكل التالى :



3 – غير الخاصية Name للمفتاح الثانى .. لتصبح القيمة CmdNext .. وهى الخاصية التى من خلالها يتم تنفيذ الكود لكى يقرأ أسم المفتاح وهو Next ..

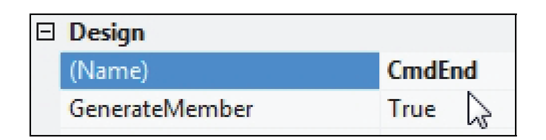


(Name)	cmdNext
GenerateMember	True



يمكن أن تكون الخاصية Text وهى الخاصة بأسم المفتاح على الشاشة ، ويمكن أن تكون باللغة العربية أو الإنجليزية .. بينما الخاصية Name لابد وأن تكون باللغة الإنجليزية حتى يتعرف الكود البرمجى عليها .. لذلك تم وضع كلمة Cmd وهى إختصار كلمة Command حتى لا يحدث تعارض بين الكلمة Next فى الخاصية Name ، والكلمة Next أيضاً فى الخاصية Text .

4 – غير الخاصية Name للمفتاح الثالث .. لتصبح القيمة CmdEnd ..

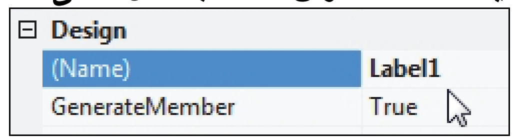


5 – إضغط على المفتاح << Next مرتين فتظهر نافذة الكود .. ثم إكتب الكود التالى :

```
Label1.Text = " دار البراء للنشر "
```



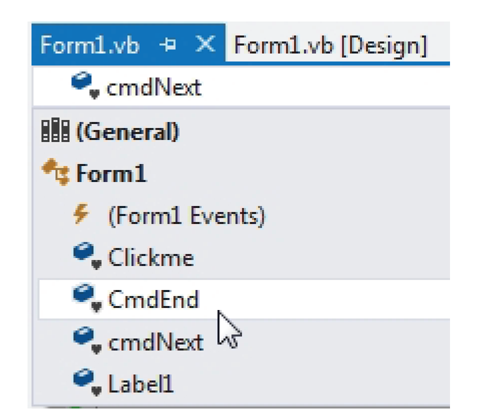
كلمة Label1 هى أسم العنوان Label وهى (Visual Basic 2013) فى الخاصية Text للعنوان .. كما بالشكل التالى :



معنى الكود السابق !!

إظهر كلمة (دار البراء للنشر) في العنوان داخل النموذج بدلاً من كلمة (Visual Basic 2013) .. وذلك عند الضغط على مفتاح << Next .

6 – من القائمة المنسدلة Object الموجودة داخل نافذة الكود .. إختار أسم المفتاح الثالث CmdEnd ..



أو بالضغط على المفتاح **End** مرتين من نافذة النموذج Form .. فتظهر نافذة الكود مرة أخرى .. ثم أكتب الكلمة ..

End

وتعنى إنهاء تنفيذ البرنامج .. فيظهر الشكل النهائى للأكواد داخل نافذة الكود .. كما بالشكل التالى :

```
Form1.vb   Form1.vb [Design]
cmdNext                              Click
Public Class Form1

    Private Sub Clickme_Click(sender As Object, e As EventArgs) Handles Clickme.Click
        MsgBox(" مرحباً بكم ق دار الراية لنشر وتوزيع الكتب والوسائط العلمية")
    End Sub

    Private Sub cmdNext_Click(sender As Object, e As EventArgs) Handles cmdNext.Click
        Label1.Text = "دار الراية لنشر"
    End Sub

    Private Sub Button2_Click(sender As Object, e As EventArgs) Handles CmdEnd.Click
        End
    End Sub
End Class
```

7 – شغل البرنامج .. بالضغط على مفتاح **F5** .



- إذا تم الضغط على مفتاح **Open** تظهر نافذة وتحتوى على رسالة .. كما بالشكل التالى :



- وإذا تم الضغط على مفتاح **Next >>** يتغير العنوان ليصبح (دار البراء للنشر) .. كما بالشكل التالى :



- أما إذا تم الضغط على مفتاح **End** .. فيتم إنهاء تنفيذ البرنامج .



يمكن إستخدام المفتاح Enter من لوحة المفاتيح لتنفيذ المفتاح النشط ، والمفتاح Esc لإنهاء وظيفة المفتاح .. كما يمكن أيضاً التنقل بين المفاتيح من خلال السهم اليمين ، والسهم الشمال الموجود بلوحة المفاتيح .

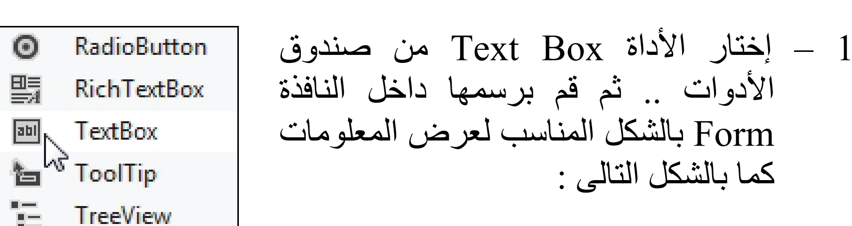
■ الأداة Text Box :

يستخدم مربع النص Text Box فى عرض معلومة معينة أثناء التشغيل .. وهذه المعلومة يتم إدخالها بواسطة المستخدم ، أو نتيجة عملية حسابية تمت داخل الكود .

والخاصية الأساسية التى تستخدم لعرض المعلومة هى النص Text ويمكن أن تكون البيانات داخل مربع النص كمدخلات Input بواسطة المستخدم ، أو مخرجات Output يظهرها لنا البرنامج عند تنفيذ خاصية أو عملية معينة .

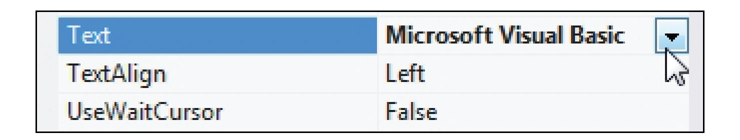
ولتنفيذ إستخدام الخاصية Text Box .. إتبع الخطوات التالية :

1 – إختار الأداة Text Box من صندوق الأدوات .. ثم قم برسمها داخل النافذة Form بالشكل المناسب لعرض المعلومات كما بالشكل التالى :

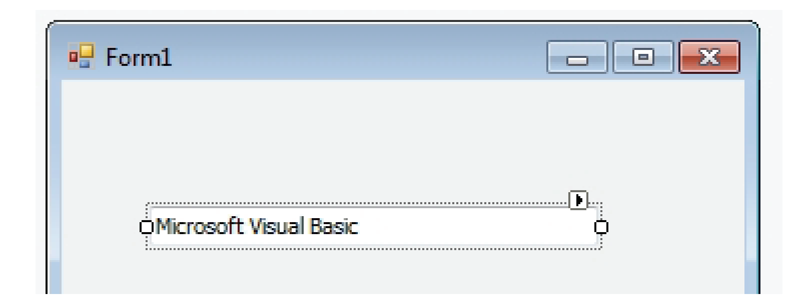




2 – أكتب النص الذى تريده داخل الخاصية Text ..

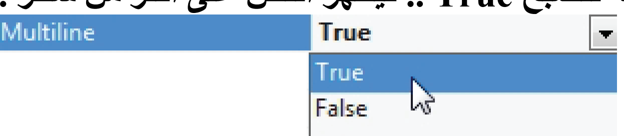


فيظهر النص داخل المستطيل فى النافذة Form .. كما بالشكل التالى :

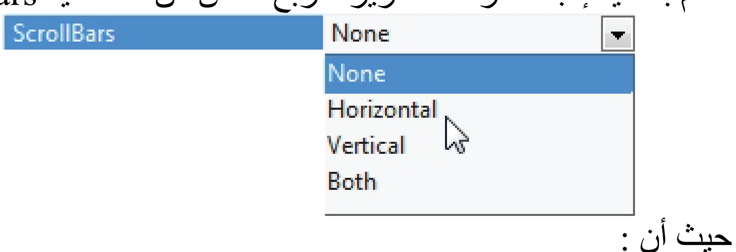




النص الذى تم إدخالة فى المستطيل Text Box .. هو نص مكون من سطر واحد فقط .. ولعرض أكثر من سطر .. إذهب للخاصية Multi line وغير القيمة لتصبح True .. فيظهر النص على أكثر من سطر ..



3 – قم بتحديد إتجاه أشرطة التمرير لمربع النص من الخاصية Scroll Bars



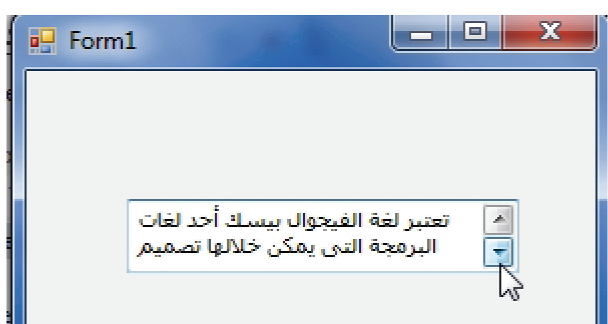
حيث أن :

• **None** .. لا يوجد أشرطة تمرير .

- **Horizontal** .. شريط تمرير أفقى .
- **Vertical** .. شريط تمرير رأسى .
- **Booth** .. شريط تمرير أفقى ورأسى معاً .

4 – إختار الخاصية **Booth** ليظهر لنا شريط تمرير أفقى ورأسى معاً .

5 – شغل البرنامج .. بالضغط على مفتاح **F5** .. فيظهر النموذج كما بالشكل التالى :

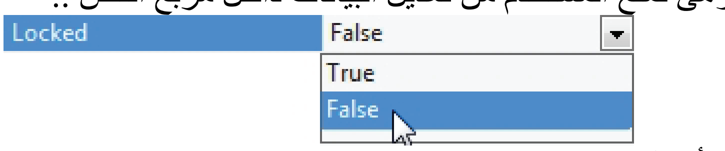


ومن النافذة السابقة .. نجد أنه يوجد سهم تمرير للنص المكتوب .. ويمكن من خلاله التحرك بالأسطر لقراءة النص المكتوب .

■ أهم خصائص مربع النص Text Box :

• الخاصية Locked :

وهى تمنع المستخدم من تعديل البيانات داخل مربع النص ..



حيث أن القيمة :

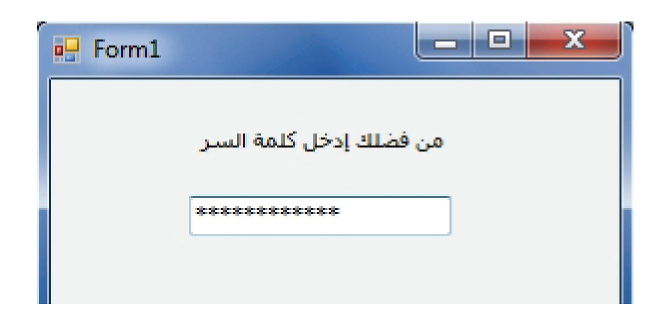
- **True** .. يمكن التعديل على النص .
- **False** .. لا يمكن التعديل على النص ، ويظل كما هو .

• **الخاصية Password Char :**

تستخدم الخاصية Password مع مربع النص Text Box .. وذلك لإظهار أى رمز بدلاً من الحروف عند إدخال كلمة السر فى مربع النص ****** ولتنفيذ ذلك .. إتبع الخطوات التالية :

1 – من نافذة الخصائص غير قيمة النص فى الخاصية Password Char وضع علامة * أو علامة ● مرة واحدة .

2 – شغل البرنامج .. بالضغط على مفتاح **F5** فتظهر نافذة البرنامج .. ثم قم بإدخال كلمة السر المناسبة فتظهر كما بالشكل التالى :

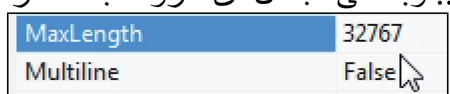


• **الخاصية Max Length :**

وهى تحدد الحد الأقصى لعدد الحروف التى يتم إدخالها فى مربع النص ، ولا يسمح بإدخال أزيد من ذلك .. ولتنفيذ ذلك إتبع الخطوات التالية :

1 – حدد مربع النص الذى يتم إدخال البيانات بداخله .

2 – من الخاصية Max Length إدخل القيمة 8 .. وهى تعنى الحد الأقصى 8 حروف فقط .. وبالتالى تجاهل أى حروف بعد الحرف الثامن ..

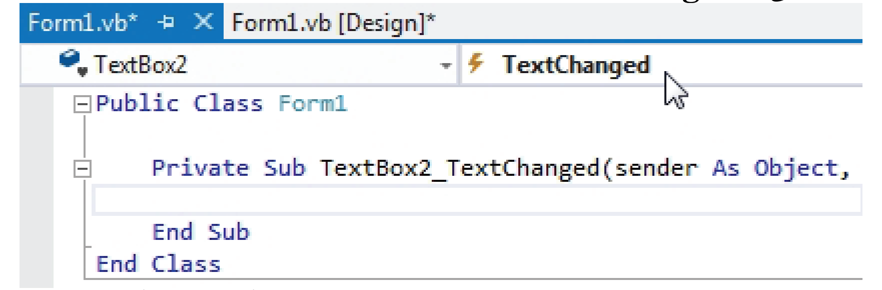




القيمة الإفتراضية للخاصية **Max Length** كحد أقصى لإدخال الحروف داخل مربع النص الواحد ..هى **32767** حرف .

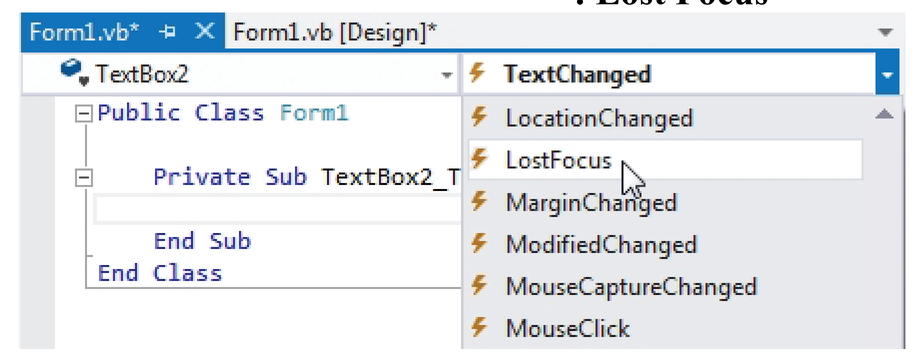


عند فتح نافذة الكود الخاصة بمربع النص **Text Box** نجد أن الحدث الإفتراضى هو **Text Change** ..



ويقوم بتغيير محتويات مربع النص .. إما بإضافة ، أو حذف أى حرف من الحروف الموجودة بداخله .

- أما الحدث **Lost Focus** :



يحدث عند إنتقال مؤشر الـ Mouse من مربع النص إلى أى عنصر أو Control أخر علـى النموذج .. وذلك بالضغط على مفتاح **Tab** من لوحة المفاتيح .

ويمكن من خلال هذا الحدث عمل قاعدة تحقق على النص الكامل الموجود داخل الـ Text Box ، والتأكد من صحة الإدخال .

• ويمكن تنفيذ طريقة عمل العنصر Text Box مع الأداة Button بالإضافة إلى خصائص كل منهما .. من خلال المثال التالى :



صمم برنامج يقوم بعمليات حسابية بسيطة مثل الجمع والطرح .. ولتنفيذ ذلك .. إتبع الخطوات التالية :

1 – إفتح مشروع جديد New Project .. كما تعلمت سابقاً .

2 – أضف عدد 3 Text Box من صندوق الأدوات داخل النموذج .

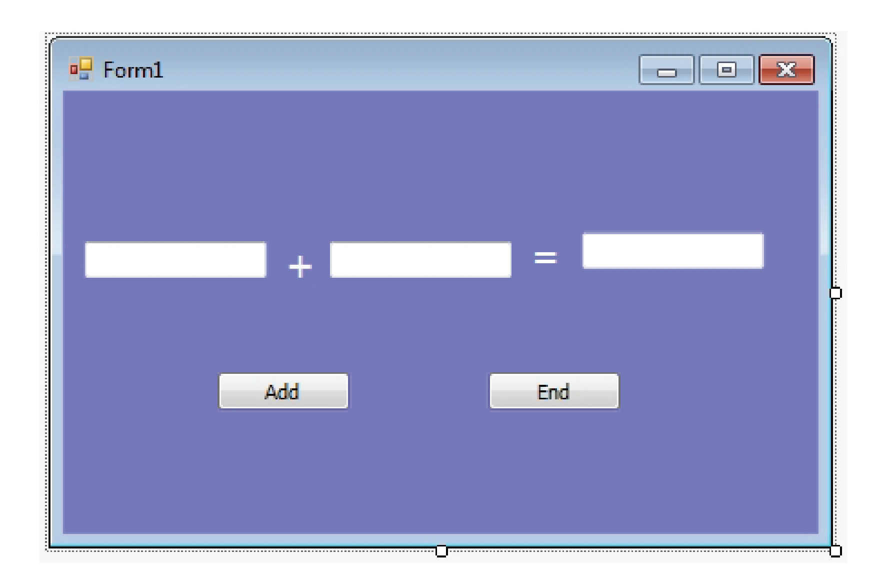
3 – أضف عدد2 مفتاح تحكم Button داخل النموذج .

4 – أضف عدد 2 Label لكتابة عنوان يظهر على النموذج .. ثم غير الخاصية Text بحيث يكون العنوان الأول علامة + ، والعنوان الثانى يكون علامة = .

5 – غير الخاصية Text للمفتاح الأول لتصبح **Add** ، والمفتاح الثانى لتصبح **End** .

6 – غير الخاصية Name للمفتاح الأول لتصبح CmdAdd ، والمفتاح الثانى لتصبح CmdEnd .. وذلك لتسهل تنفيذ الكود البرمجى .

7 – قم بعمل التنسيقات اللازمة لكل من الـ Text Box ، و الـ Buttons ، والـ Labels .. فيظهر النموذج كما بالشكل التالى :



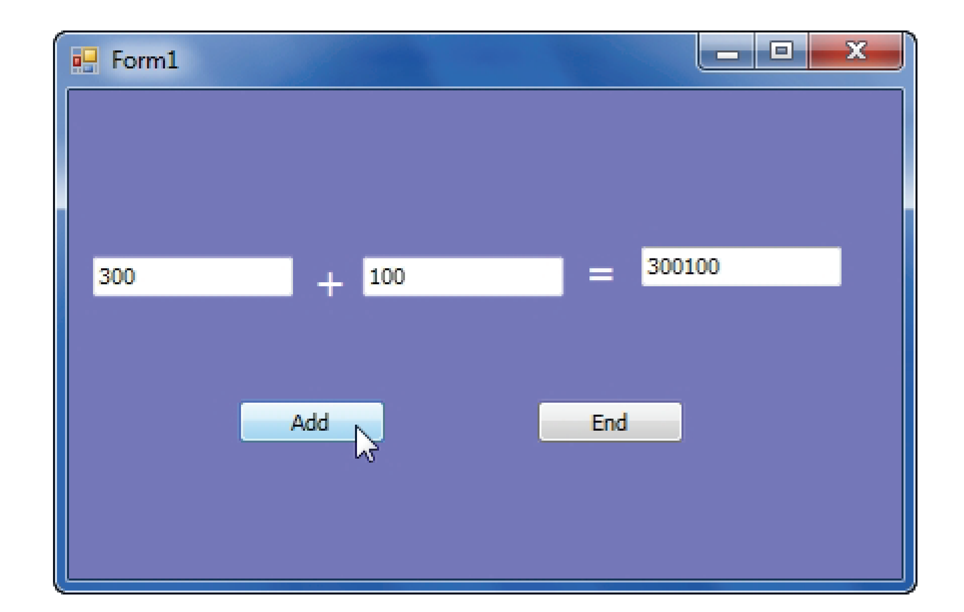
8 – إضغط على المفتاح **Add** مرتين فتظهر نافذة الأكواد .. ثم أكتب الكود التالى :

```
TextBox3.Text=TextBox1.Text+TextBox2.Text
```

معنى الكود السابق !!

أن مربع النص الثالث TextBox3 = مربع النص الأول TextBox1 مع إدخال قيمة داخلة + مربع النص الثانى TextBox2 مع إدخال قيمة داخلة .. سوف تظهر النتيجة فى مربع النص الثالث عند الضغط على المفتاح Add .

9 – شغل البرنامج .. بالضغط على مفتاح **F5** .. وقم بإجراء عملية حسابية كما بالشكل التالى :





بعد إجراء العملية الحسابية فى الخطوة السابقة قام البرنامج بجمع القيم كحروف وليس أرقام .. وهذا بالطبع خطأ .. لذلك لابد من وضع الدالة Val وهى إختصار Value فى الكود البرمجى السابق .. لتقوم بتحويل النص Text إلى رقم Number .. فيظهر الكود بعد وضع الدالة كما بالشكل التالى :

```
TextBox3.Text = Val (TextBox1.Text) + Val (TextBox2.Text)
```



تـم وضـع قـوسين لـجملة مربع الـنص الأول ، ومـربع الـنص الثانى بـعد كلمة Val .

معنى الكود السابق !!

أننا نقول لجهاز الكمبيوتر .. قم بتحويل القيمة التى تحتويها الخاصية Text لمربع النص الأول والثانى إلى رقم .. ثم قم بجمعهم ووضع الناتج فى مربع النص الثالث .

10 – إضغط على المفتاح **End** مرتين .. ثم قم بكتابة جملة الإنهاء التالية الخاصة بالمفتاح :



ويكون الشكل النهائى لنافذة الكود .. كما بالشكل التالى :

```
(General)                                   (Declarations)
Public Class Form1

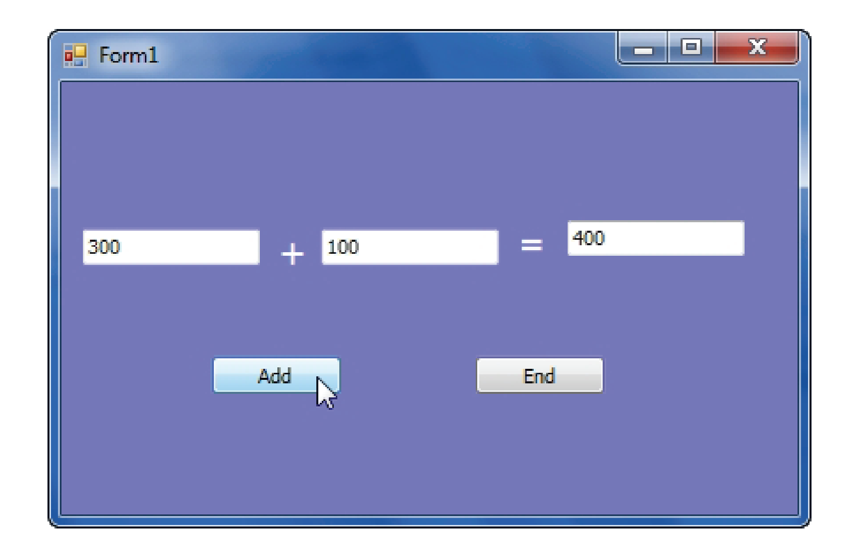
    Private Sub TextBox2_TextChanged(sender As Object, e As EventArgs) Handles TextBox2.TextChanged

    End Sub

    Private Sub CmdAdd_Click(sender As Object, e As EventArgs) Handles CmdAdd.Click
        TextBox3.Text = Val(TextBox1.Text) + Val(TextBox2.Text)
    End Sub

    Private Sub CmdEnd_Click(sender As Object, e As EventArgs) Handles CmdEnd.Click
        End
    End Sub
End Class
```

11 – شغل البرنامج .. بالضغط على مفتاح **F5** .. ثم قم بإجراء عملية حسابية وإضغط على المفتاح **Add** .. لتظهر النتيجة .. كما بالشكل التالى :



■ **الأداة Radio Button :**

تستخدم فى عرض إختيار يمكن وضعه بالقيم إما (**Yes / No**) ، أو (**On / Off**) .. وغالباً ما يستخدم هذا العنصر مجموعة من الإختيارات والتى يمكن خلالها أن يختار المستخدم إختيار أو Option واحد فقط .. مثل المؤهل الدراسى (دكتورة / ماچستير / بكالوريوس / ليسانس) ، أو الحالة الإجتماعية (أعزب / متزوج) .

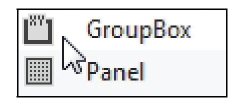


لعمل مجموعة من الإختيارات Options داخل النموذج .. يتم وضع الإختيارات داخل وعاء يسمى Group Box .

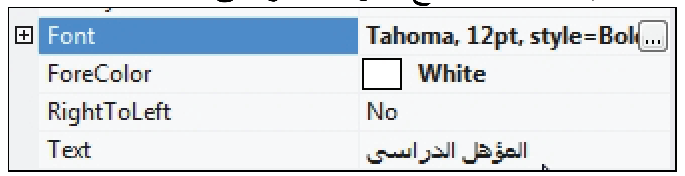
ولتنفيذ إستخدام الأداة Group Box مع الأداة Radio Button .. إتبع الخطوات التالية :

1 – أضف نموذج جديد للمشروع .

2 – أضف الأداة Group Box من صندوق الأدوات Tool Box ..



3 – غير الخاصية Text لتصبح المؤهل الدراسى ..



ثم غير لون الخط من الخاصية Fore Color ، نوع وحجم الخط من الخاصية Font .. فيظهر الوعاء كما بالشكل التالى :

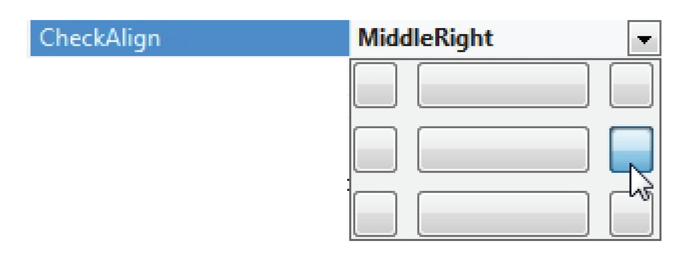


4 – أضف عدد 4 Radio Box من صندوق الأدوات Tool Box ..

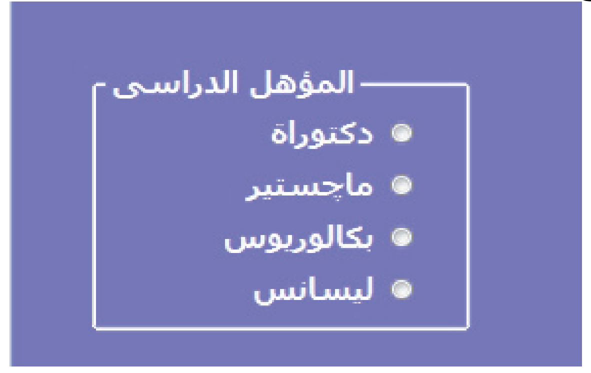


5 – غير الخاصية Text لكل إختيار ليصبح الإختيار الأول (دكتوراة) ، والإختيار الثانى (ماجستير) ، والإختيار الثالث (بكالوريوس) ، والإختيار الرابع (ليسانس) .

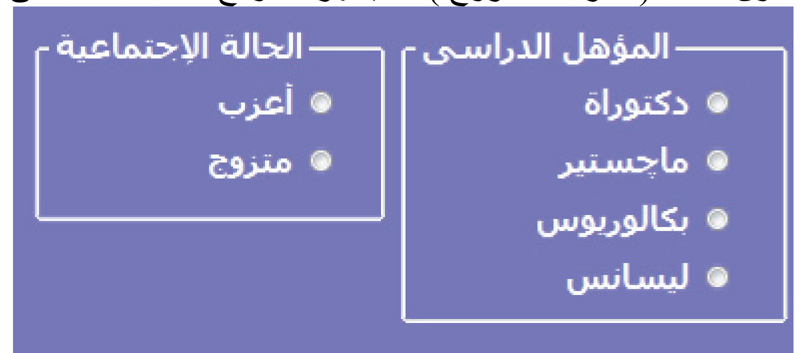
6 – غير الخاصية Check Align للإختيارات الأربعة الموجودة داخل النموذج لتكون Middle Right .. وذلك لتظهر على اليمين لأن النص المكتوب باللغة العربية ..



فيظهر النموذج .. كما بالشكل التالى :



7 – كرر الخطوات السابقة .. ثم قم بإضافة إختيارات الحالة الإجتماعية لتكون (أعزب / متزوج) .. فيظهر النموذج كما بالشكل التالى :



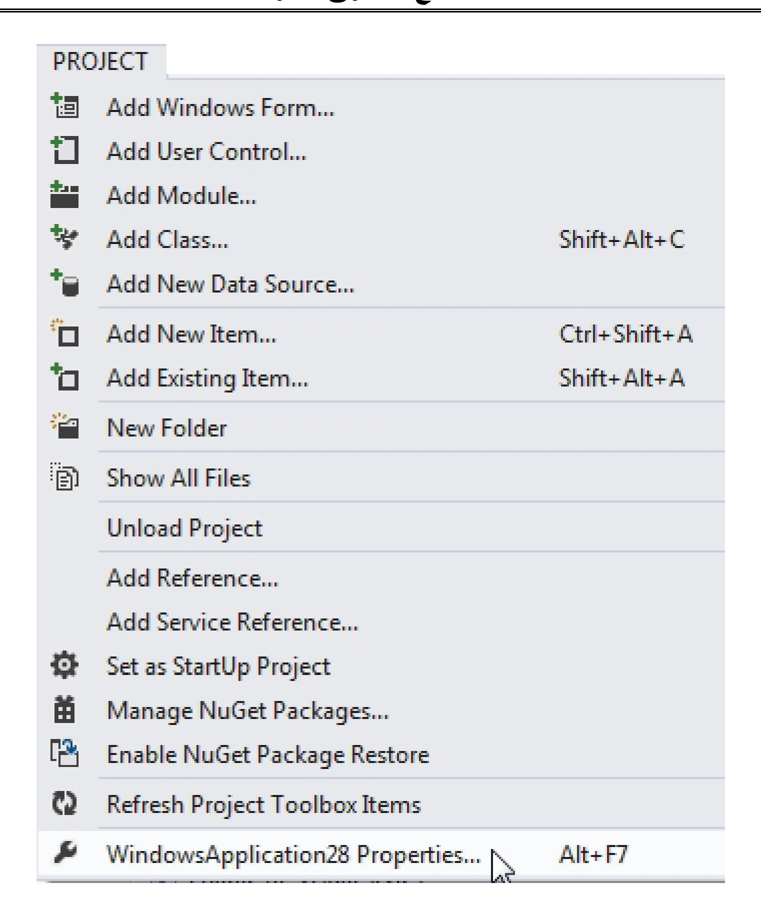


عند تحريك الوعاء الذى تحتوى بداخلة مجموعة الإختيارات .. سوف تتحرك معه تلقائياً مجموعة الإختيارات المكونة للوعاء .

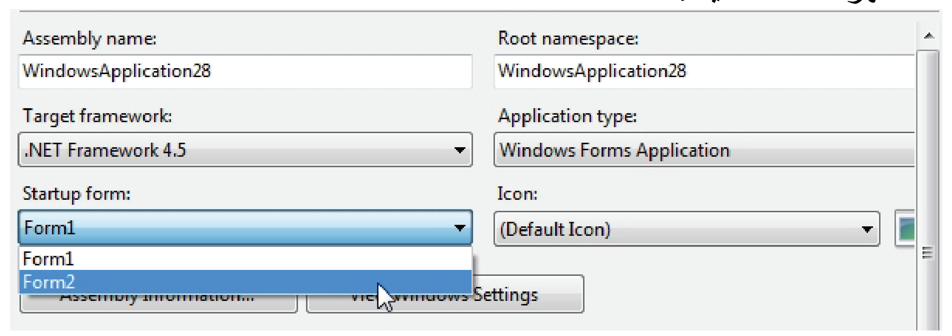


فى حالة وجود أكثر من نموذج Form داخل المشروع .. عند تنفيذ البرنامج يقوم تلقائياً البرنامج بتشغيل Form 1 .. أما إذا أردنا تغير نافذة بدء البرنامج لتصبح Form 2 .. إتبع الخطوات التالية :

- من القائمة Project إختار الأمر Windows Application Properties أو بالضغط على مفتاحى **Alt + F7** ..

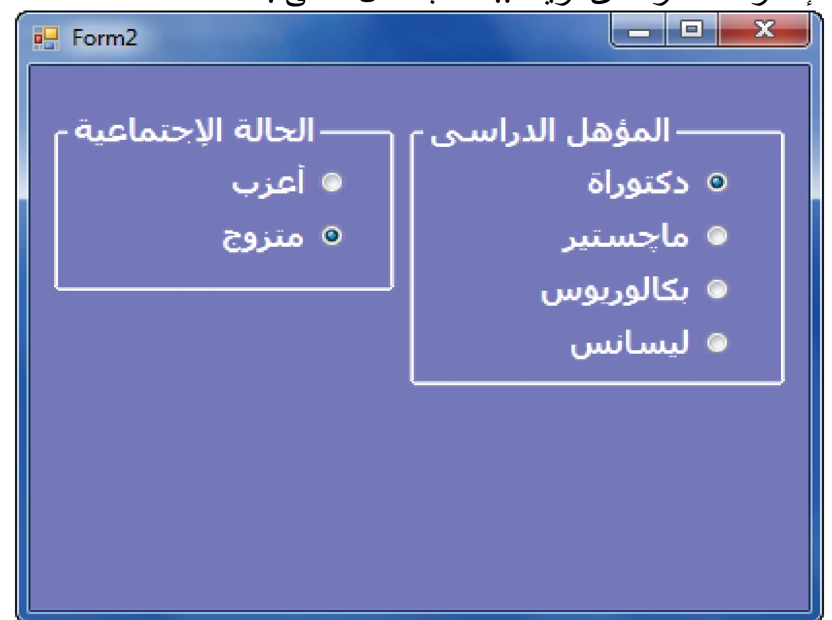


- فتظهر النافذة التالية :



- إختار من القائمة المنسدلة Startup Form النموذج المطلوب إظهاره فى البداية .. وهو Form 2 .

8 – شغل البرنامج .. بالضغط على مفتاح **F5** .. فتظهر نافذة البرنامج .. ثم إختار العنصر الذى تريده .. كما بالشكل التالى :



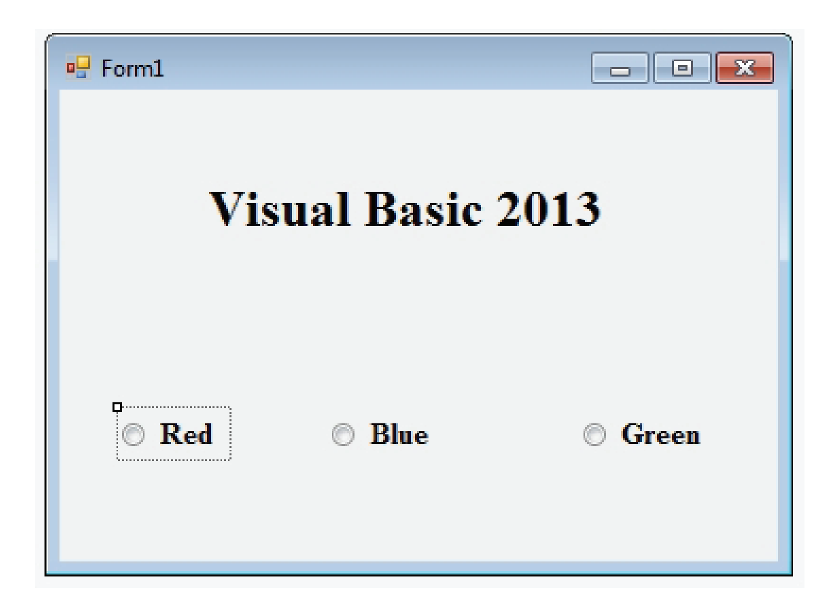


صمم برنامج يقوم بتغيير لون الخط بمجرد تنشيط أسم اللون على النافذة .. ولتنفيذ ذلك .. إتبع الخطوات التالية :

1 – إفتح مشروع جديد New Project .

2 – أضف عنوان Label من صندوق الأدوات Tool Box ، ثم غير الخاصية Text ليصبح النص (**Visual Basic 2013**) .

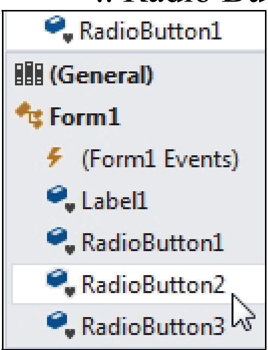
3 – أضف عدد 3 Radio Button من صندوق الأدوات Tool Box ، ثم غير الخاصية Text للأداة الأولى لتصبح اللون Red ، والأداة الثانية لتصبح Blue ، والأداة الثالثة لتصبح Green .. فيظهر النموذج كما بالشكل التالى :



4 – إضغط مرتين على الأداة Radio Button 1 المسمى Red .. فتظهر نافذة الكود .. ثم أكتب الكود التالى :

```
Label1.ForeColor=Color.Red
```

5 – إفتح القائمة المنسدلة Object الموجودة داخل نافذة الكود .. ثم إختار أداة الإختيار الثانية Radio Button 2 ..



ثم إختار الحدث Click من القائمة المنسدلة Procedure .



يمكن الضغط مرتين على الأداة Radio Button 2 من النافذة Form فتظهر نافذة الكود .. بدلاً من إختيار الأداة من القائمة Object أو إختيار الحدث من القائمة Procedure .. وذلك لتسهيل عملية كتابة الكود حسب رغبة المستخدم .

6 – أكتب الكود التالى :

```
Label1.ForeColor=Color.Blue
```

7 – كرر الخطوتين السابقتين للأداة Radio Button 3 .. ثم أكتب الكود التالى :

```
Label1.ForeColor=Color.Green
```

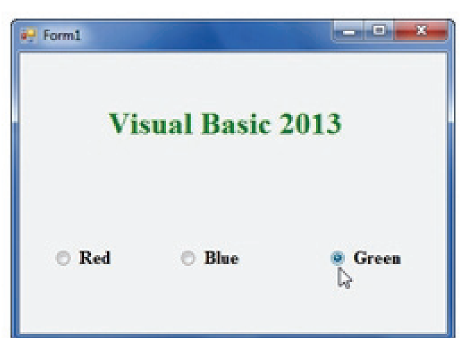


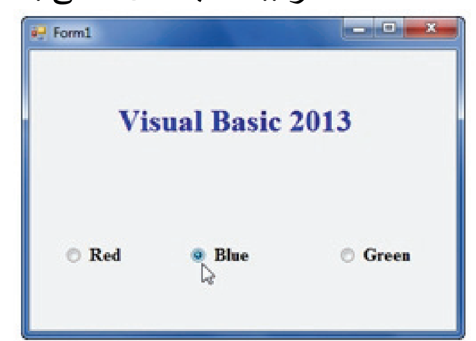
يمكن نسخ الكود لكل Check Box بدلاً من إعادة كتابتة مرة أخرى .. ثم غير أسم اللون فقط .

8 – شغل البرنامج .. بالضغط على مفتاح **F5** ..فتظهر نافذة البرنامج كما بالشكل التالى :



9 – نشط الإختيار **Red** ليتغير لون النص إلى اللون الأحمر .. وبتنشيط الإختيار **Blue** يتغير لون النص إلى اللون الأزرق .. وهكذا للون الأخضر .. كما بالشكل التالى :





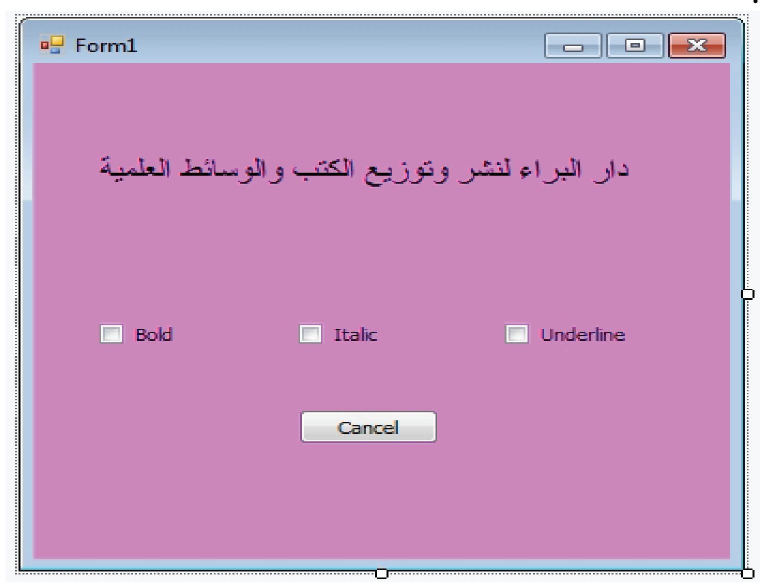
■ الأداة Check Box ☑ :

هى أداة لعمل إختيارات أيضاً .. وذلك بوضع علامة داخل مربع الإختيار .. والأداة Check Box تشبه الأداة Radio Button فى جميع الخصائص .. ولكن مع إختلاف رئيسى هو أنه يمكن إختيار أكثر من عنصر فى وقت واحد على عكس العنصر Radio Buttons الذى يسمح بإختيار عنصر واحد فقط .

ويتم تنفيذ أداء عمل العنصر Check Buttons بناءاً على شرط معين يتم كتابتة داخل البرنامج .



صمم برنامج يقوم بعمل تغيير لسمك الخط للنص المكتوب .. كما بالشكل التالى :



ولتنفيذ ذلك إتبع الخطوات التالية :

1 – إفتح مشروع جديد New Project .

2 – أضف عنوان Label ، ثم أضف عدد 3 Check Box ، وإضافة مفتاح للتحكم Button من صندوق الأدوات Tool Box .

3 – غير الخاصية Text للعنوان Label 1 لتصبح (دار البراء لنشر وتوزيع الكتب والوسائط العلمية) .

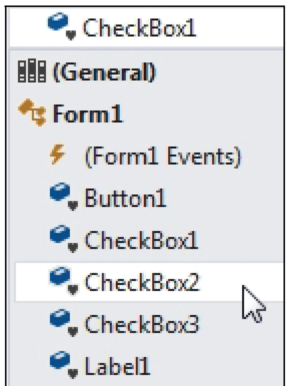
4 – غير الخاصية Text لـ Check Box الأول لتصبح (Bold) أى خط ثقيل ، والثانى لتصبح (Italic) أى خط مائل ، والثالث لتصبح (Underline) أى خط مسطر .

5 – غير الخاصية Text للمفتاح Button لتصبح Cancel .

6 – إضغط مرتين على الأداة Check Box المسمى Bold .. فتظهر نافذة الكود .. ثم أكتب الكود التالى :

```
Label1.Font=New Font(Label1.Font, FontStyle.Bold)
```

7 – إفتح القائمة المنسدلة Object الموجودة بنافذة الكود .. ثـم إخـتار الأداة Check Box 2 ..



ثم إختار الحدث Click من القائمة المنسدلة Procedure .. ثم أكتب الكود التالى :

```
Label1.Font=New Font(Label1.Font, FontStyle.Italic)
```

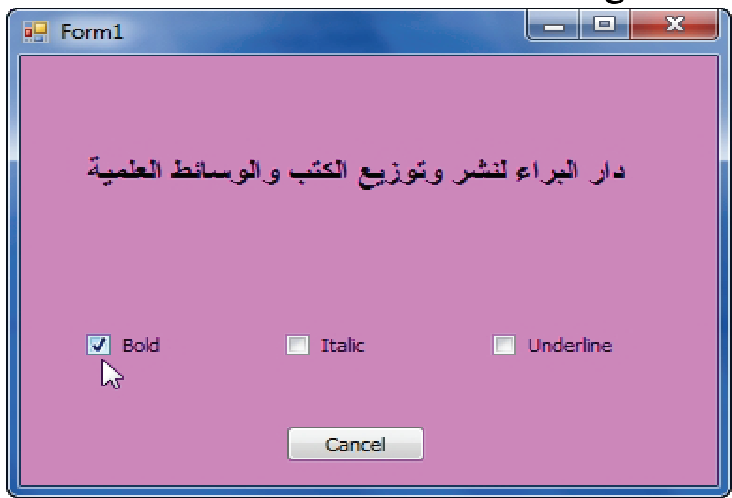
8 – كرر الخطوة السابقة مرة أخرى مع الإداة Check Box 3 .. أو إضغط مرتين على الأداة Check Box 3 من النافذة Form .. ثم أكتب الكود التالى :

```
Label1.Font=New Font(Label1.Font, FontStyle.Underline)
```

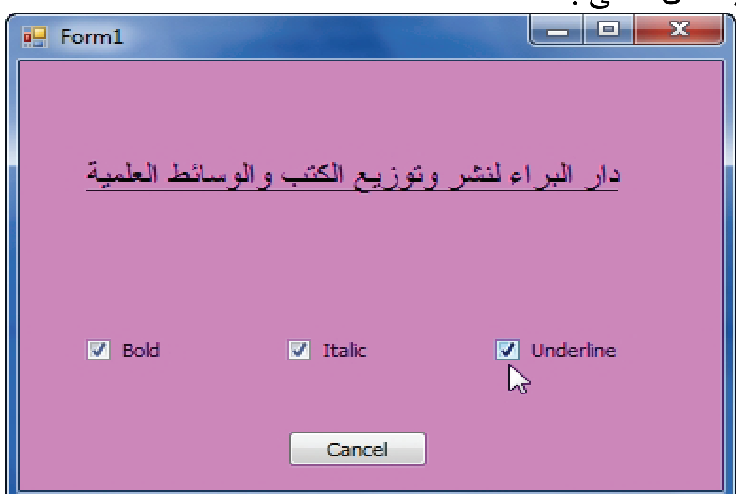
9 – إختار المفتاح Button 1 من القائمة المنسدلة Object ، وإختار الحدث Click من القائمة Procedure .. وأكتب الجملة :

End

10– شغل البرنامج .. بالضغط على مفتاح **F5** .. فيظهر نموذج البرنامج كما بالشكل التالى :



11 – نشط الإختيار Bold ليتحول العنوان إلى خط ثقيل .. كما بالشكل السابق ، وإذا تم تنشيط الإختيار Italic تحول العنوان إلى خط مائل ، وإذ تم تنشيط الإختيار Under Line تحول الخط إلى خط مسطر .. كما بالشكل التالى :



12 – إضغط مفتاح **Cancel** إنهاء تشغيل البرنامج .

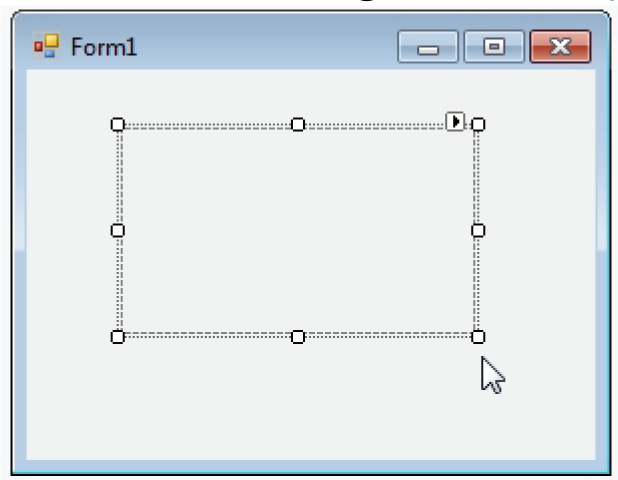
■ الأداة Picture Box :

يستخدم لعرض الصور والرسوم داخل النموذج Form .. والخاصية المستخدمة فى ذلك هى الخاصية Image من نافذة الخصائص .

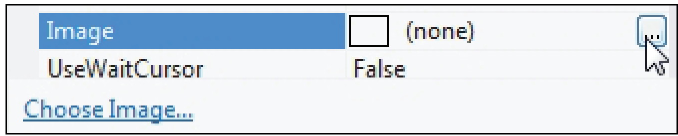
ولتنفيذ إستخدام الأداة Picture Box .. إتبع الخطوات التالية :

1 – إفتح مشروع جديد .. ثم أضف نموذج جديد New Form للمشروع الحالى .

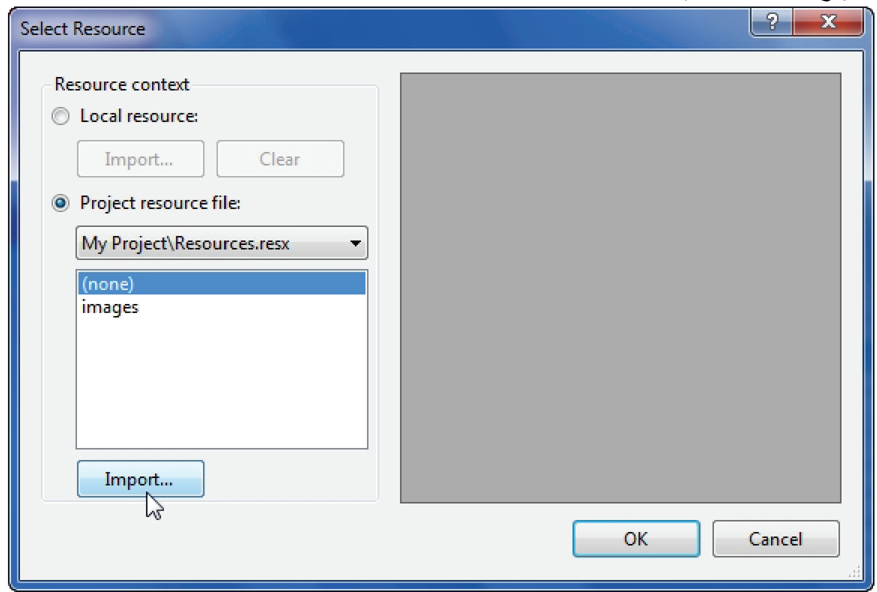
2 – إختار الأداة Picture Box من صندوق الأدوات Tool Box .. ثم قم بضبط العنصر داخل النموذج Form بالمقاس المناسب لحجم الصورة المختارة .. كما بالشكل التالى :



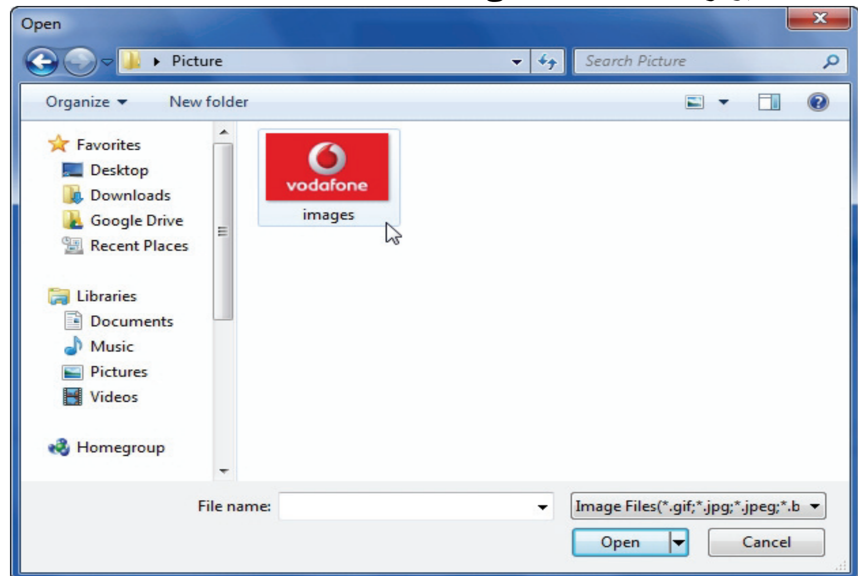
3 – إضغط على المربع الموجود يمين الخاصية Image من نافذة الخصائص ..



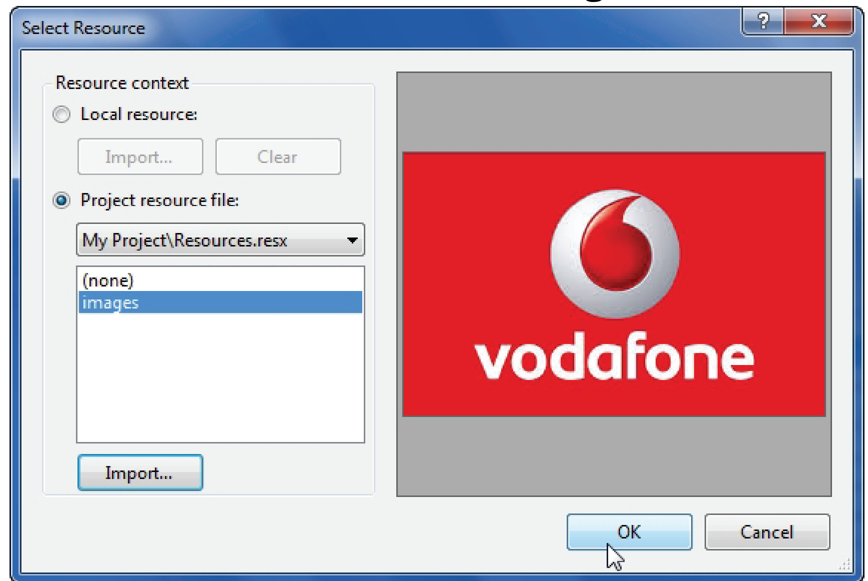
فتظهر النافذة التالية :



4 – إضغط **Import** .. فتظهر نافذة لتحديد مكان الصورة على جهاز الكمبيوتر كما بالشكل التالى :



5 – إختار الصورة .. ثم إضغط **Open** .. لتظهر الصورة فى نافذة الإختيار .. كما بالشكل التالى :



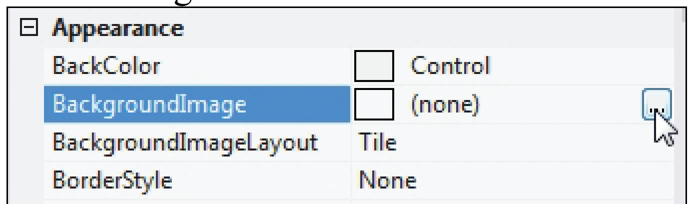
6 – إضغط **OK** .. فتظهر الصورة فى نافذة النموذج Form .. كما بالشكل التالى :



■ إضافة صورة لخلفية النموذج Background Image :

يمكن إضافة صورة لخلفية النموذج Form بدلاً من إستخدام الألوان التقليدية وذاك بالمقاس المناسب حسب حجم الـ Form .. ولتنفيذ ذلك إتبع الخطوات التالية :

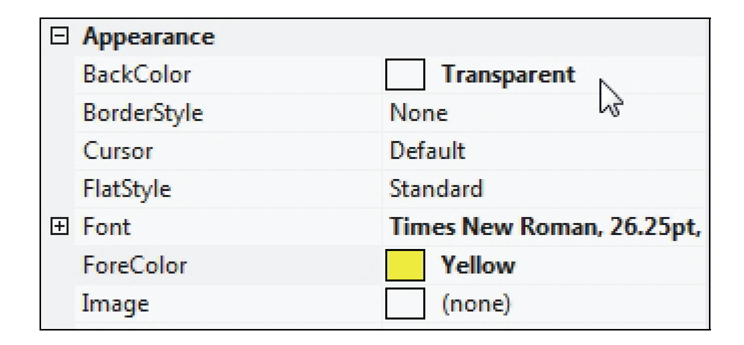
1 – إضغط على المربع [...] الموجود يمين الخاصية ..
Back Ground Image



2 – فتظهر نافذة تحديد مكان الصور على جهاز الكمبيوتر .. إختار الصورة المناسبة ثم إضغط **OK** فتظهر الخلفية كما بالشكل التالى :



3 – إختار الأداة Label من صندوق الأدوات Tool Box .. ثم غير الخاصية Back Color لتكون Transparentأى خلفية النص شفافة ..



4 – أكتب العنوان المناسب للنموذج فى الخاصية Text .. فيظهر الشكل النهائى لـ Form ليمثل واجهة تطبيق البرنامج .. كما بالشكل التالى :



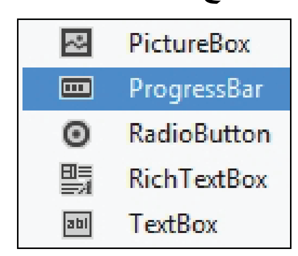


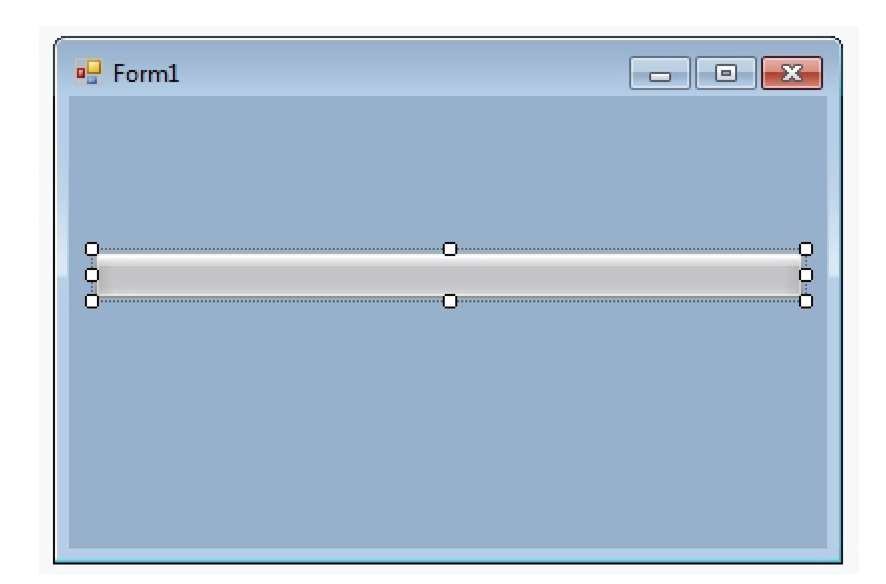
يمكن إضافة الصور المتحركة بإمتداد GIF للنموذج Form .. لتظهر بشكل متحرك أثناء تشغيل البرنامج لعمل جاذبية أثناء إستخدام البرنامج .

■ الأداة Progress Bar :

تستخدم الأداة Progress Bar لعمل شريط إنتظار عند بدء تشغيل البرنامج ولتنفيذ طريقة إستخدام الـ Progress Bar .. إتبع الخطوات التالية :

1 – أضف الأداة Progress Bar من صندوق الأدوات Tool Box إلى النموذج Form .. فتظهر الأداة داخل النموذج .. كما بالشكل التالي :

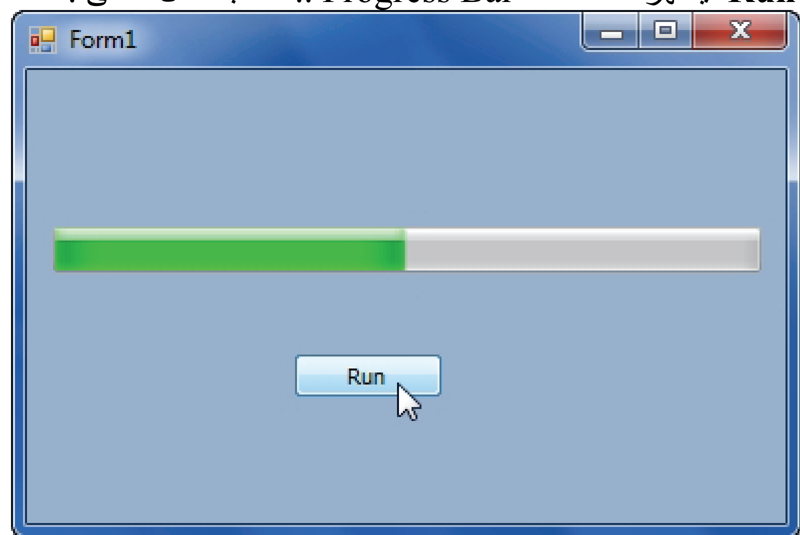




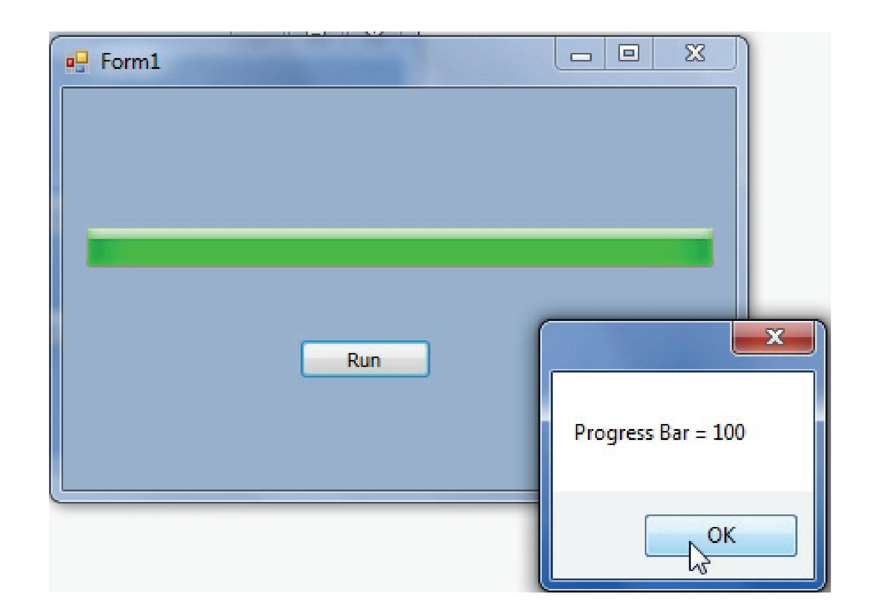
2 – أضف مفتاح للتحكم Button للنموذج .. ثم غير أسم المفتاح من الخاصية Text ليكون **Run** .. ثم إضغط مرتين على المفتاح **Run** فتظهر نافذة الكود .. ثم أكتب الكود التالى :

```
IfProgressBar1.Value = 100 Then
    MessageBox.Show("Progress Bar = 100")
Else
    ProgressBar1.Value = ProgressBar1.Value + 10
End If
```

3 – شغل البرنامج .. بالضغط على مفتاح **F5** .. ثم إضغط على المفتاح **Run** فيظهر أداء الأداة Progress Bar .. كما بالشكل التالى :



4 – كرر الضغط على المفتاح **Run** حتى يصل شريط التحميل إلى النهاية فتظهر الرسالة التالية :





يمكن التحكم فى عدد مرات التحميل وسرعة الضغط على المفتاح Run وذلك بزيادة القيمة Value فى الكود السابق لتصبح مثلاً 50 بدلاً من 10 .

■ الأداة List Box :

هى أداة تستخدم لعرض قائمة لمجموعة من البنود فى شكل قائمة .. لكى يستطيع المستخدم إختيار بند أو أكثر منها .. ولها مجموعة من الخصائص التى تساعد على تحديد العناصر أو البند الذى يتم إختياره .

ومن أهم هذه الخصائص الخاصية Items .. وهى تمكن المستخدم من تعبئة القائمة بالعناصر.. ويتم ذلك من خلال كتابة كود يحدد رقم العنصر داخل القائمة وإعطاؤه القيمة المطلوبة .

ولتنفيذ إستخدام الأداة List Box .. إتبع الخطوات التالية :

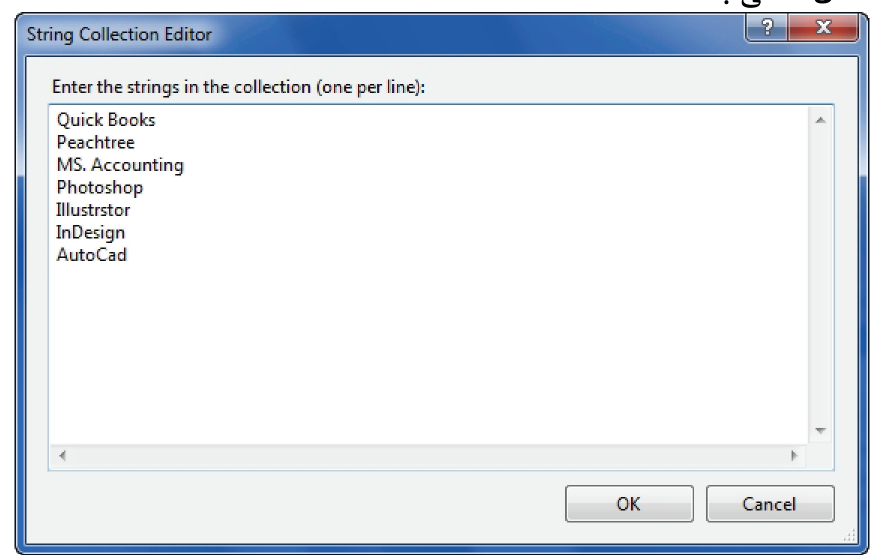
1 – إفتح مشروع جديد New Project .. كما تعلمت فى الأدوات السابقة .

2 – إختار الأداة List Box من صندوق الأدوات Text Box داخل النموذج Form .. ثم قم بضبط الأداة لتظهر بالشكل المناسب عند تشغيل البرنامج .

3 – إضغط على المربع [...] الموجود يمين الخاصية Items ..



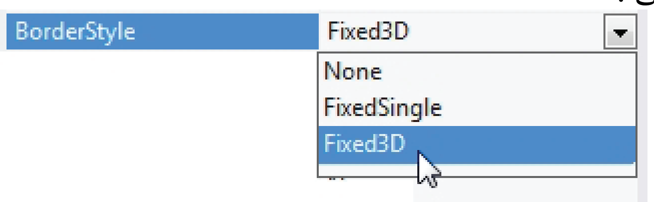
فتظهر النافذة String Collection Editor .. ثم أضف بنود القائمة كما بالشكل التالى :



4 – إضغط **OK** .. فتظهر القائمة داخل النموذج كما بالشكل التالى :



5 – إختار شكل إطار القائمة من الخاصية Border Style .. كما بالشكل التالى :



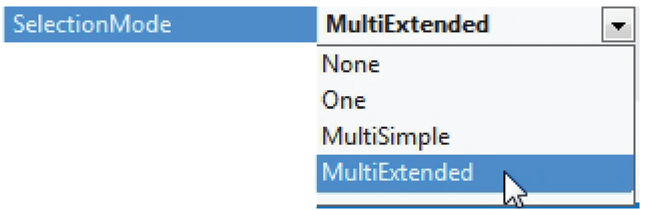
6 – شغل البرنامج .. بالضغط على مفتاح **F5** .. فتظهر نافذة البرنامج كما بالشكل التالى :



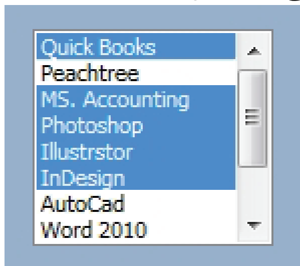
فتظهرالقائمة وبها شريط تمرير رأسى Scroll Bar .. وذلك إختيار أى عنصر من القائمة .



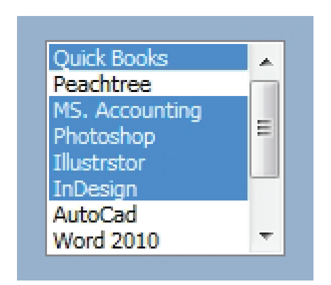
القائمة السابقة يتم إختيار عنصر واحد منها فقط .. أما إذا أرادنا إختيار أكثر من عنصر معاً فى وقت واحد .. يتم تغيير قيمة الخاصية Selection Mode ..



- **One** .. إختيار عنصر واحد فقط .
- **Multi Simple** .. إختيارأكثر من عنصر معاً واحداً يلى الأخر .. بمجرد الضغط بالـ Mouse على العنصر .. كما بالشكل التالى :



- **Multi Extend** .. إختيار أكثر من عنصر معاً فى نفس الوقت .. وذلك بالضغط على مفتاح **Ctrl** من لوحة المفاتيح أثناء الإختيار .. كما بالشكل التالى :



■ **الأداة Combo Box :**

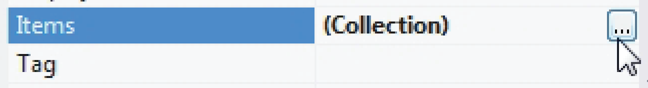
وهى تشبه إلى حد ما الأداة List Box .. ولكن تختلف فى الشكل حيث تظهر فى شكل قائمة منسدلة وبها العناصر ، ولا يمكن إختيار إلا عنصر واحد فقط من القائمة .. ولتنفيذ إستخدام الأداة Combo Box .. إتبع الخطوات التالية :

1 – إفتح مشروع جديد ، ثم أضف الأداة Conbo Box من صندوق الأدوات Tool Box .. كما بالشكل التالى :

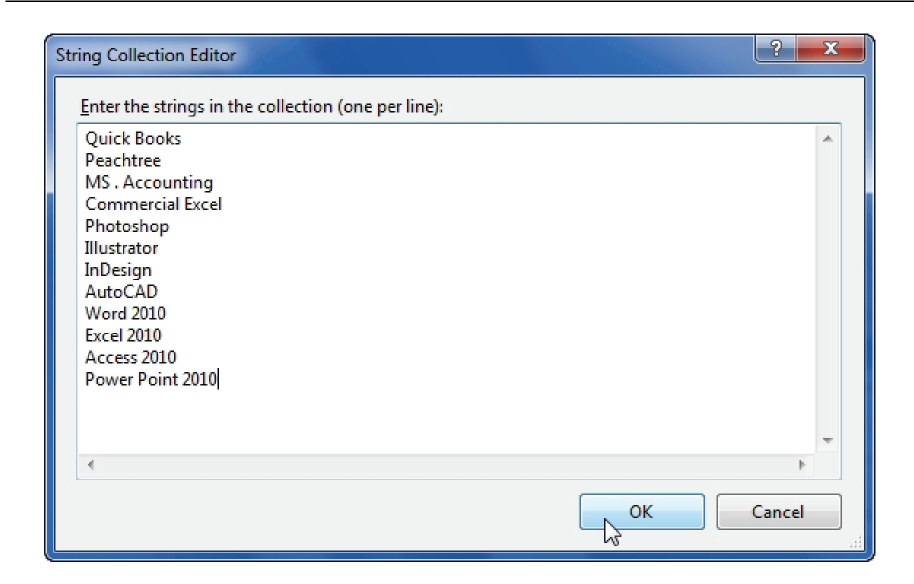


2 – غير الخاصية Text .. لتصبح كلمة (إختار الكتاب) .

3 – إضغط على المربع الموجود يمين الخاصية Items ..



فتظهر نافذة إدخال البيانات التالية :



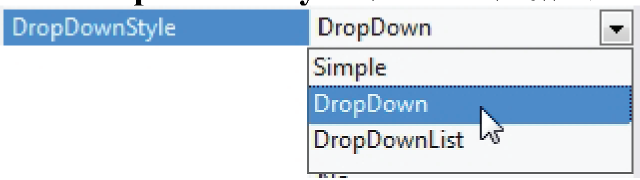
4 – إدخل بيانات القائمة .. ثم إضغط **OK** .

5 – شغل البرنامج بالضغط على مفتاح **F5** .. فتظهر نافذة البرنامج .. كما بالشكل التالى :



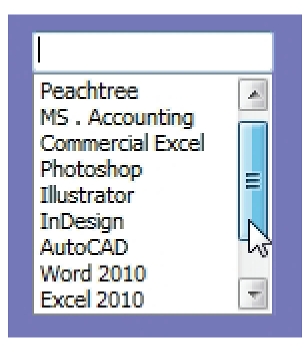


يمكن التحكم فى شكل طريقة عرض القائمة **Combo Box** داخل البرنامج وذلك بتغيير قيمة الخاصية **Drop Down Style** ..

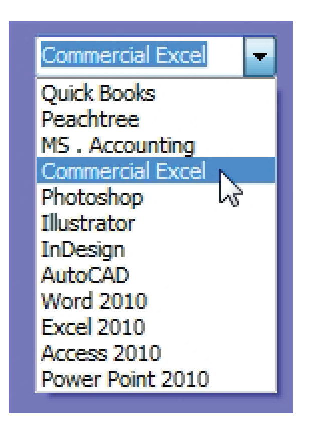


حيث أن :

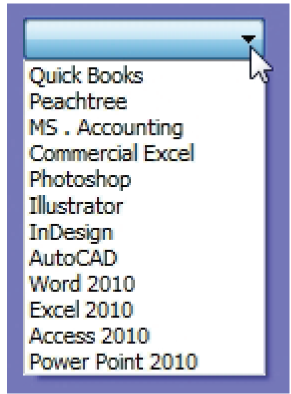
- **Simple** .. تظهر العناصر داخل القائمة من خلال شريط تمرير رأسى .. كما بالشكل التالى :



- **Drop Down** .. تظهر العناصر فى شكل قائمة منسدلة .. بمجرد كتابة أول حرف من العنصر المطلوب يتحدد تلقائياً داخل القائمة .. كما بالشكل التالى :



- **Drop Down List** .. تظهر القائمة فى شكل مفتاح تحكم .. بمجرد الضغط عليه تظهر عناصر القائمة .. كما بالشكل التالى :



■ **الأداة Timer** :

تستخدم فى تنفيذ كود محدد خلال فترة زمنية معينة .. وهو من العناصر التى لاتظهر أثناء تشغيل البرنامج .. ولكن تظهر نتيجة أدائها اثناء التشغيل .

ولتنفيذ إستخدام الأداة Timer من خلال المثال التالى :



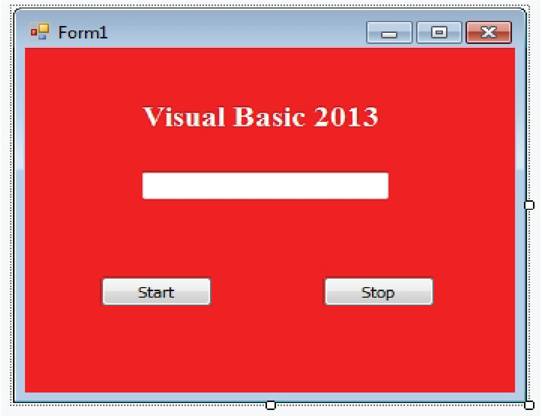
صمم برنامج يقوم كل ثانيتين بزيادة القيمة الموجودة فى مربع إدخال القيم بمقدار رقم 1 .. ولتنفيذ ذلك .. إتبع الخطوات التالية :

1 – إفتح مشروع جديد .. كما تعلمت سابقاً .. ثم غير لون الخلفية باللون المناسب .

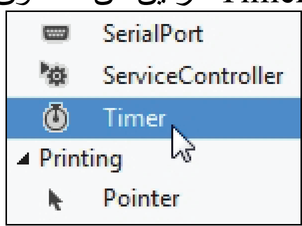
2 – أضف عنوان Label .. ثم غير أسم العنوان من الخاصية Text .

3 – أضف الأداة Text Box للنموذج Form .

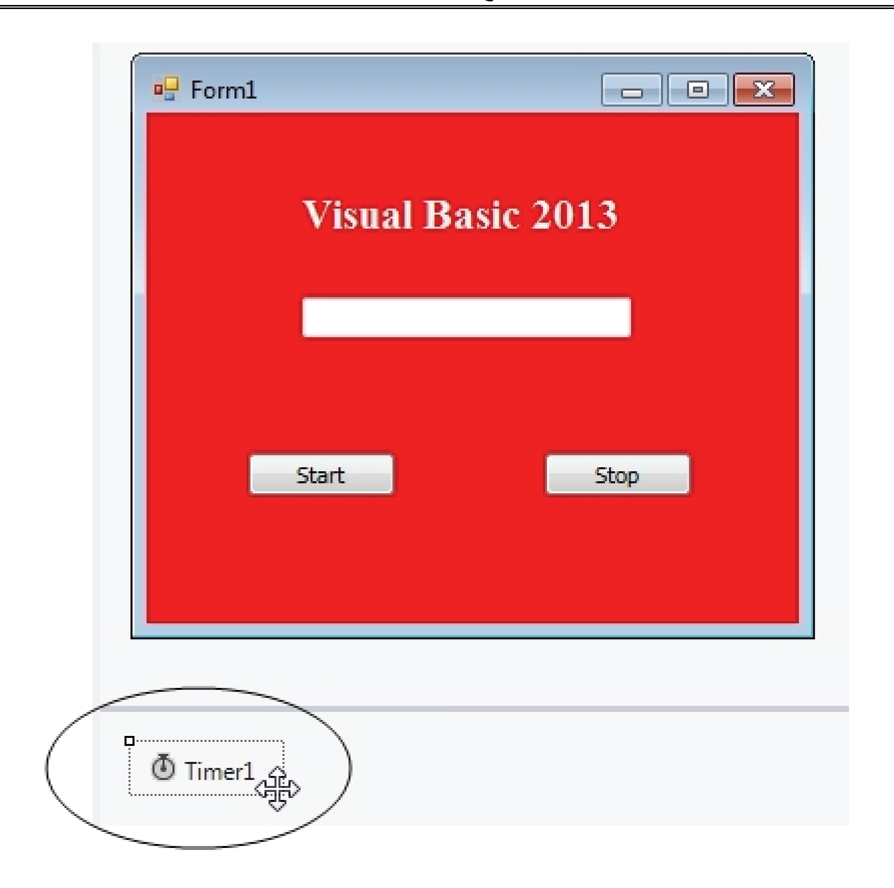
4 – أضف عدد 2 مفتاح تحكم Button .. وغير أسم المفتاح من الخاصية Text .. ليكون أسم المفتاح الأول Start ، والثانى Stop .. فيظهر النموذج .. كما بالشكل التالى :



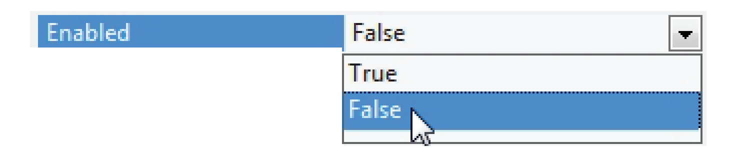
5 – إضغط على الأداة Timer مرتين من صندوق الأدوات Tool Box ..



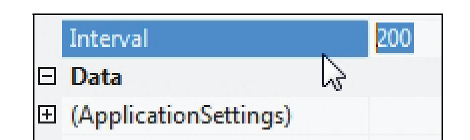
فتظهر الأداة Timer أسفل نافذة النموذج .. كما بالشكل التالى :



6 – إضغط على الأداة Timer الموجود بالنافذة السابقة .. ثم من نافذة الخصائص Properties .. إختار القيمة False من الخاصية Enabled وهى عدم تفعيل الأداة عند تشغيل البرنامج ، إلا بعد إدخال القيمة ..



7 – حدد التوقيت الزمنى للحركة أثناء التشغيل .. وذلك من الخاصية Interval





الثانية الواحدة للحركة داخل الخاصية Interval تساوى قيمة 1000 .. فإذا أرادنا الحركة بعد 10 ثوانى نكتب القيمة 10000 .

8 – أكتب القيمة 200 .. وذلك لسرعة الحركة الخاصة بالـ Timer أثناء تشغيل البرنامج .

9 – إضغط مرتين على المفتاح **Start** .. فتظهر نافذة الكود .. ثم أكتب الكود التالى :

```
Timer1.Start
```

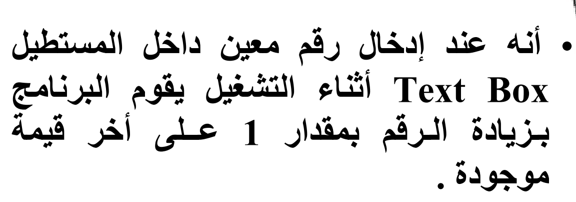
10 – إضغط مرتين على المفتاح **Stop** .. ثم أكتب الكود التالى :

```
Timer1.Stop
```

11 – إضغط مرتين على الأداة **Timer** .. فتظهر نافذة الكود .. ليتم كتابة الكود الخاص بتنفيذ المهمة عند الضغط على المفتاح **Start** ، أو المفتاح **Stop** كما يلى :

```
TextBox1.Text=TextBox1.Text+1
```

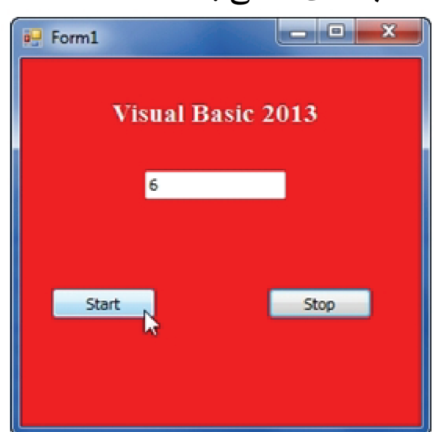


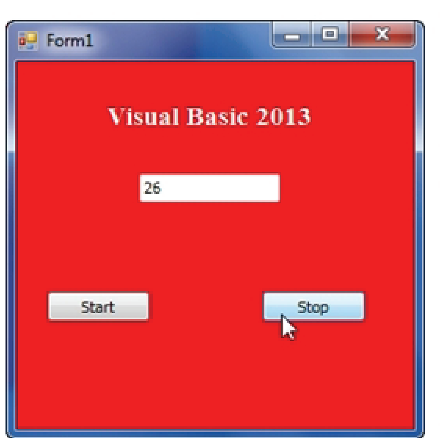


• أنه عند إدخال رقم معين داخل المستطيل Text Box أثناء التشغيل يقوم البرنامج بزيادة الرقم بمقدار 1 علـى أخر قيمة موجودة .

12– شغل البرنامج .. بالضغط على مفتاح **F5** .

13 – إدخل أى قيمة رقمية ، وليكن الرقم 5 .. داخل المستطيل Text Box ، ثم إضغط مفتاح **Start** .. فيقوم البرنامج بزيادة القيمة بمقدار 1 .. كما بالشكل التالى :





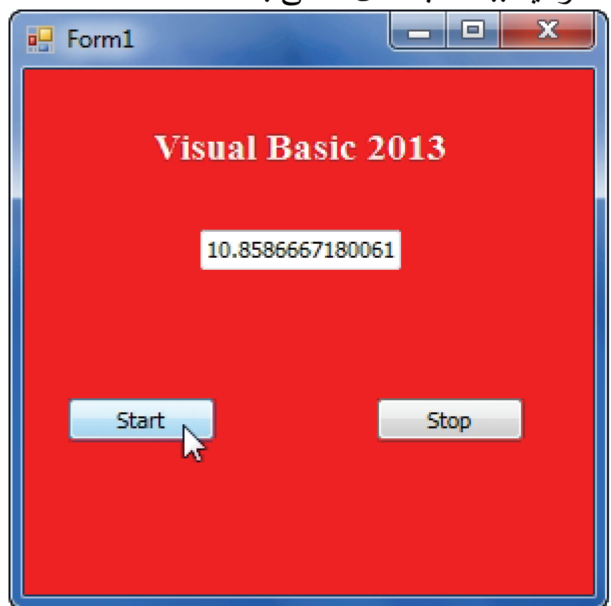
14 – إضغط مفتاح **Stop** لإيقاف تنفيذ تشغيل البرنامج .



لتوليد أرقام عشوائية بدلاً من أرقام ثابتة .. قم بإضافة الدالة Rnd .. وهى إختصار Random داخل الكود الخاص بالـ Timer .. فيظهر الكود كما بالشكل التالى :

```
TextBox1.Text=TextBox1.Text+Rnd
```

شغل البرنامج **F5** .. وقم بإدخال الرقم 1 .. ثم إضغط مفتاح **Start** .. فتظهر أرقام عشوائية .. كما بالشكل التالى :

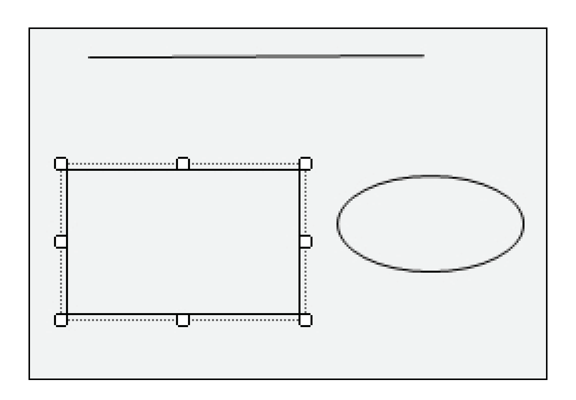


■ أدوات الـ Shape :

LineShape
OvalShape
RectangleShape

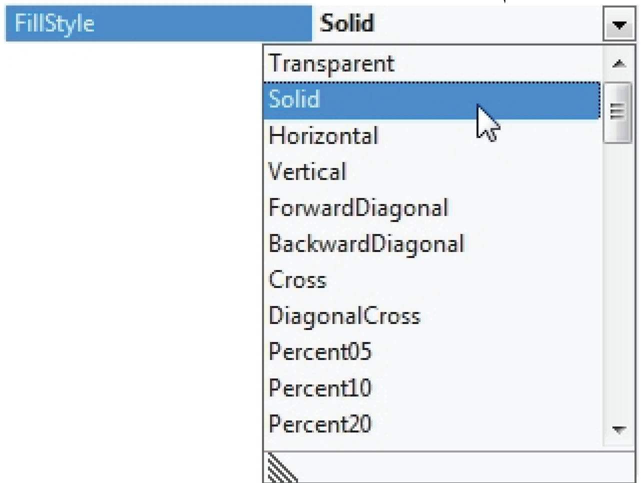
تستخدم لرسم عناصر الجرافيكس على النموذج ، وهى تعطى شكل جمالى للتصميم .. وتساعد على الفصل بين العناصر الكثيرة داخل النموذج الواحد .

ولتنفيذ إستخدام أدوات الـ Shape .. إضغط مرتين على الشكل المناسب من صندوق الأدوات Tool Box .. فتظهر الأشكال كما بالشكل التالى :

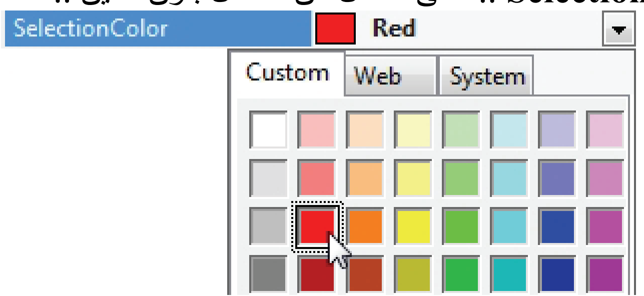


• ولتغيير خصائص الأشكال Shape .. حدد الشكل المطلوب تغيير خصائصة ، ثم من نافذة الخصائص Propereties إختار :

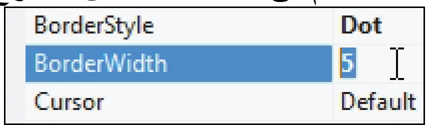
• **Fill Style** .. للتحكم فى شكل ملئ اللون الداخلى للشكل ..



• **Selection Color** .. لملئ الشكل من الداخل بلون معين ..

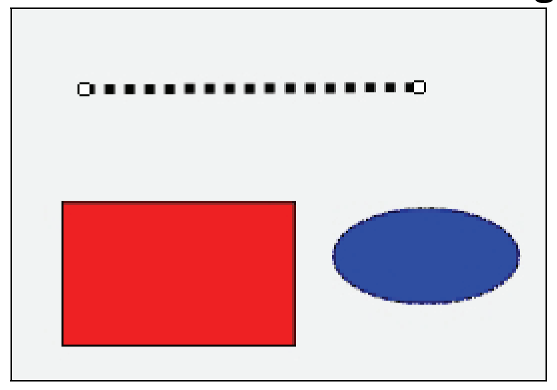


• **Border Style** .. للتحكم فى إطار الشكل من الخارج .
• **Border Width** .. للتحكم فى سمك الشكل من الخارج .



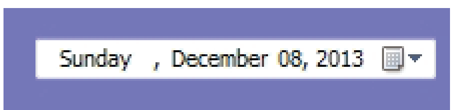
BorderStyle	Dot
BorderWidth	5
Cursor	Default

وبعد تطبيق الخصائص السابقة على الأشكال Shapes .. تظهر الأشكال كما بالشكل التالى :



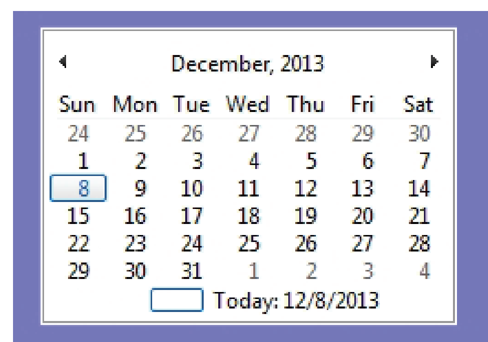
■ **الأداة Date Time Picker** :

تستخدم لإضافة تاريخ اليوم للنموذج Form .. إضغط مرتين على الأداة Date Time Picker من صندوق الأدوات Tool Box .. فـتظهر كما بالشكل التالى :



■ **الأداة Month Calander** :

تستخدم لإضافة نتيجة الشهر الحالى للنموذج Form .. إضغط مرتين على الأداة Month Calander من صندوق الأدوات Tool Box .. فتظهر كما بالشكل التالى :





صمم برنامج لإظهار تاريخ اليوم ، ونتيجة الشهر الحالى .. ولتنفيذ ذلك إتبع الخطوات التالية :

1 – إفتح مشروع جديد .. ثم أضف عنوان Label ، ثم غير الخاصية Text لتكون 2014 .

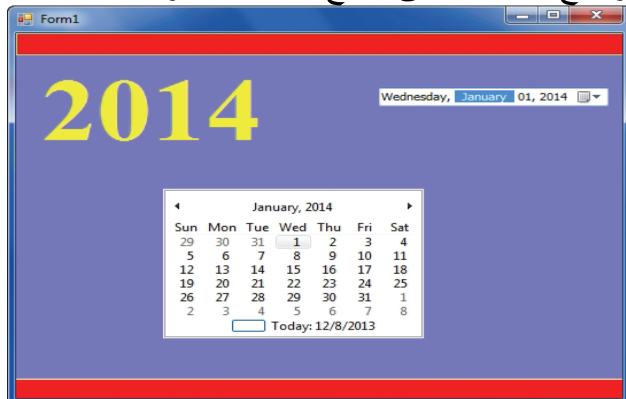
2 – أضف الأداة Date Time Picker .. لتعبرعن تاريخ اليوم .

3 – أضف الأداة Month Calander .. لتعبر عن نتيجة الشهر الحالى .

4 – أرسم مستطيلين أعلى وأسفل النموذج بإستخدام الأداة Rectangule ، ثم غير خاصية اللون لتكون اللون الأحمر .

5 – قم بعمل التنسيقات اللازمة بالشكل المناسب للنموذج .

6 – شغل البرنامج بالضغط على مفتاح **F5** .. فيظهر كما بالشكل التالى :



الفصل السادس

المتغيرات والثوابت

Variable and Constants

نتعلم فى هذا الفصل ..

- تعريف المتغيرات Variable .
- شروط المتغيرات (البيانات) .
- أنواع المتغيرات (المحلية / العامة) .
- تعريف الثوابت Constants .
- أمثلة على المتغيرات والثوابت .
- التعليقات Comments .

الفصل السادس

المتغيرات والثوابت

Variable and Constants

■ المتغيرات Variables :

• **المتغيرات** ، **أو البيانات** .. هى جمل تكتب داخل الإجراءات Procedures .. بحيث تكون كل جملة ، أو أمر فى سطر منفصل .. وإذا كان هناك أكثر من جملة فى الكود الواحد تنفذ كل جملة واحدة تلو الأخرى بنفس الترتيب .

• ومن الممكن أيضاً كتابة أكثر من أمر فى نفس السطر .. وذلك بوضع علامة : نقتطين بين الجملة ، والجملة التالية لتصبح سطر واحد فقط بدلاً من سطرين .

• كما يمكن أيضاً تقسيم الأمر الواحد على أكثر من سطر .. وذلك بوضع علامة _ Underscore فى نهاية السطر .. ثم الضغط على مفتاح **Enter** لإستكمال باقى الجملة فى السطر التالى .. وبذلك يصبح الأمر مقسم على سطرين .

• **المتغيرات** .. تتغير قيمتها عند إدخال البيانات بواسطة المستخدم .. مثل حقل الأسم ، إما أن يكون باللغة العربية ، أو يتغير للغة الإنجليزية .. وبالتالى يستطيع البرنامج التعرف على الكلمة المكتوبة .

• **المتغير** .. هو مكان فى الذاكرة يتم حجزه وتخصيصة لإستخدام قيمة معينة من المستخدم أو من البرنامج نفسه .

• يجب أن يكون المتغير أسم ونوع .. كما يجب أيضاً أن نعلن عنه قبل إستخدامه ونستخدم كلمة **Dim** للإستعلام عن المتغيرات .

▪ شروط المتغيرات (البيانات) :

1 – ألا يزيد عدد الحروف عن 255 حرف .
2 – يجب أن يبدأ بحرف وليس رقم .. وهذا الحرف من الممكن أن يكون حرف كبير Capital ، أو حرف صغير Small .

3 – كما يجب ألا يحتوى على مسافات خالية ، أو نقطة ، أو أى حرف خاص بخلاف علامة _ Underscore .

4 – يجب ألا يسمى المتغير بأسم أو كلمة محجوزة تستخدم فى لغة الـ Visual Basic .. مثل كلمة For ، أو IF .

• فمثلاً .. يمكن كتابة جملة المتغير داخل نافذة الكود .. كما بالصورة التالية :

```
Dim distance As Integer
```

• أما إذا تم كتابة المتغير بشكل خاطئ .. أى ترك مسافات بين الكلمتين .. كما بالصورة التالية :

```
Dim User age As Integer
```

سوف تظهر رسالة خطأ .. لوجود مسافة بين الكلمة User ، والكلمة age .

• ولعلاج مشكلة المسافة بين الكلمتين لابد من وضع علامة _ Underscore بين الكلمتين .. لتصبح كما بالصورة التالية :

```
Dim User_age As Integer
```



من الممكن تعريف أكثر من جملة داخل المتغير Dim فى نفس السطر .. وذلك بوضع علامة , ثم الإعلان عن أسم المتغير التالى .. مثل :

```
Dim distance As Integer,depth As Integer
```

■ أنواع المتغيرات (البيانات) :

• أولاً : المتغيرات المحلية Local Variable :

هى المتغيرات التى لا يراها إلا الـ Procedure Form Load فقط ، لأنها تكتب داخله .. ولن يراها أى إجراء Procedure أخر فى البرنامج .. وتستخدم هذه المتغيرات كلمة **Dim** .

وتحتفظ هذه المتغيرات بقيمتها طوال فترة تشغيل الإجراء Procedure ، وبمجرد الإنتهاء منه تفقد قيمتها كلها .. وتنقسم المتغيرات المحلية Local إلى :

1 – المتغيرات الرقمية :

وهى المتغيرات التى تتعامل مع الأرقام فقط .. وتنقسم إلى :

• Integer :

متغير يقبل القيمة الموجبة 32768 إلى 32767 – ، ويشغل مساحة من الذاكرة بحجم 2 Byte .

• Long :

متغير يقبل أعداد صحيحة تتعدى 2 مليار ، ويشغل مساحة 4 Byte من الذاكرة وللإعلان عنه تاخذ الجملة الصورة التالية :

```
Dim Lgname As Long
```

حيث تعبر كلمة Lgname عن أسم المتغير .

• Single :

متغير للتعامل الأرقام الحقيقية ذات العلامات العشرية ، ويشغل مساحة 4 بايت من الذاكرة .. وللإعلان عنه تاخذ الجملة الصورة التالية :

```
Dim Sgname As Sigle
```

• Double :

متغير للتعامل مع الأرقام ذات العلامات العشرية .. ولكن كبيرة ، ويشغل مساحة 8 Byte من الذاكرة .. وللإعلان عنه تاخذ الجملة الصورة التالية :

```
Dim Dbname As Double
```

2 – المتغير String :

ويتعامل مع النصوص .. ويستقبل حتى 65500 حرف ، وكل حرف يشغل مساحة 1 Byte من الذاكرة .. وللإعلان عنه تاخذ الجملة الصورة التالية :

```
Dim Stname As String
```



• إذا أرادنا وضع عدد معين من الحروف ، أو طول معين للمتغير النصى نكتب علامة * بعد كلمة String .. ثم كتابة عدد الحروف اللازمة .. مثل الجملة التالية :

```
Dim Stname As String * 5
```

• أما إذا أعطينا المتغير نص يتعدى الـ 5 حروف يأخذ منها أول 5 حروف فقط ويتم تجاهل باقى الحروف .

3 – المتغير Currency :

ويستخدم للتعبير عن العملات ، ويشغل مساحة 8 Byte من الذاكرة .. وللإعلان عنه تاخذ الجملة الصورة التالية :

```
Dim Crname As Currency
```



يتميز المتغير Currency بخاصية عدم التقريب .. مما يعطى دقة عند حساب القيمة المالية .

3 – المتغير Date :

يستخدم للتعبير عن التاريخ والوقت ، ويشغل مساحة 8 Byte من الذاكرة .. وللإعلان عنه تاخذ الجملة الصورة التالية :

```
Dim Dtname As Date
```

4 – المتغير Boolean :

يستخدم لتخزين قيمة منطقية أما **True** ، أو **False** ، ويشغل مساحة 2 Byte من الذاكرة .. وللإعلان عنه تاخذ الجملة الصورة التالية :

```
Dim Blname As Boolean
```

5 – المتغير Byte :

يستخدم للتعبير عن عدد صحيح محصور بين صفر ، و - 255 ، ويشغل مساحة 1 Byte من الذاكرة .. وللإعلان عنه تاخذ الجملة الصورة التالية :

```
Dim Bytename As Byte
```

6 – المتغير Variant :

ويقبل أى نوع من أنواع المتغيرات السابقة ، ويشغل مساحة 16 Byte من الذاكرة .. وللإعلان عنه تاخذ الجملة الصورة التالية :

```
Dim Vrname As Variant
```



- **يمكن كتابة الجملة بدول كلمة As Variant .. فيقوم البرنامج تلقائياً بفهم الجملة لتصبح :**

```
Dim Vrname
```

- **يعتبر أى نوع من المتغيرات لا يذكر نوعه داخل لغة الـ Visual Basic على أنه Variant .**

7 – المتغير Object :

يشير هذا النوع إلى أحد الكائنات الموجودة داخل الـ Visual Basic ، وللإعلان عنه تاخذ الجملة الصورة التالية :

```
Dim Objcom As CommandButton
```



فى بعض الأحيان نحتاج إستخدام متغير يراه جميع الإجراءات Procedures على نفس النموذج Form ، أو الـ Module .. ويسمى هذا بـ Form Wide ، أو Module Wide .

ولتنفيذ الإعلان إتبع الخطوات التالية :

1 – من القائمة المنسدلة Object الموجودة داخل نافذة الكود.. إختار الأمر General .. فتظهر فى القائمة Procedure كلمة Declarnation .. كما بالشكل التالى :

```
Form1.vb*  Form1.vb [Design]*
(General)                    (Declarations)
Public Class Form1

    Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        Dim vrname
    End Sub
End Class
```

2 – أكتب جملة أى متغير من المتغيرات السابقة .. ثم إضغط **Enter** .. فيظهر خط يفصل هذا المتغير عن باقى الـ Procedure .



لابد أن ينتبه المبرمج أثناء كتابة كود البرنامج إلى قيمة هذا المتغير .

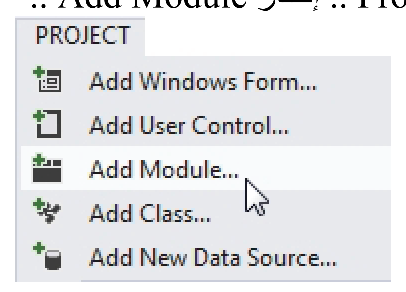
• ثانياً : المتغيرات العامة Public Variable :

وهى المتغيرات التى يراها جميع الإجراءات Procedure ، والملفات داخل النظام System .. وهو ما يسمى أيضاً بالـ Globle ، ونستخدم معه كلمة **Public** بدلاً من كلمة **Dim** .. ويتم كتابة إما فى منطقة الـ ..General Declarnations

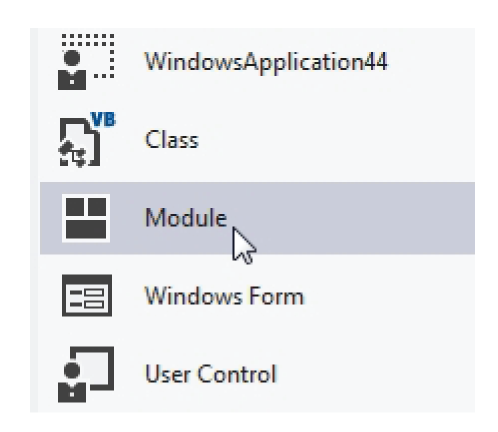
أو فى ميديول Module منفصل .

ولتنفيذ إستخدام المتغيرات العامة Public .. إتبع الخطوات التالية :

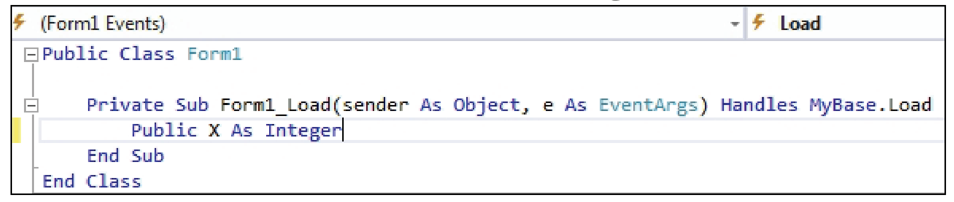
1 – من القائمة Project .. إختار Add Module ..



فتظهر النافذة التالية :



2 – إختار منها Module .. ثم إضغط مفتاح **Add** .. فتظهر نافذة كتابة الكود.. كما بالشكل التالى :



3 – أكتب الجملة التالية :





حيث أن .. المتغير X متغير عددى ويمثل رقم صحيح وهو الـ Globle وتراه جميع إجراءات البرنامج داخل الـ Procedure بلا إستثناء .. ويحتفظ بقيمتة طوال تشغيل البرنامج .

4 – إغلق نافذة الكود الخاصة بالـ Module ، والعودة لنافذة الكود الخاصة بالنموذج Form .



يحتفظ الـ Visual Basic بقيمة المتغير حتى الإنتهاء من عمل الـ procedure .. ويتم ذلك بأن نكتب كلمة Static بدلاً من كلمة Dim .

5 – قم بالإعلان عن متغير جديد بالقيمة **Z** .. كما بالصورة التالية :

```
Public Z As Integer
```

• حيث أن المتغير **Z** يأخذ قيمة ، وهذه القيمة ستتغير طوال فترة تشغيل الـ Procedure Form Load .

• **فالسؤال هنا ..**

ماذا يحدث لو أخطأنا فى كتابة أسم متغير ؟

- **فمثلاً** .. لو تم كتابة كلمة **dept** بدلاً من كلمة **Depth** .
- **هنا** .. عند تشغيل البرنامج لم يظهر عنوان هذه الكلمة .

• **إذن :**

يتضح لنا مما سبق .. أننا لا نلتزم بتعريف المتغير قبل إستخدامه ، ولكن هذا قد يسبب مشاكل كثيرة عند تنفيذ البرنامج .

• **ولتفادى ذلك الخطأ :**

إذهب إلى منطقة الإعلان عن المتغيرات ، وقبل إستخدام جملة **Dim X** أكتب كلمة Option Explicit .. كما بالصورة التالية :

```
Option Explicit Dim X
```



تأثير هذه الجملة يطبق على المتغيرات التى تكتب بعد إضافة هذه الجملة .

■ الثوابت Constants:

الثوابت فهى تشبه المتغيرات تماماً .. فلابد أن يكون للثابت أسم ونوع أيضاً ، ولكن الثوابت تحتفظ بقيمة ثابتة .. وليست متغيرة كما فى المتغيرات .

ويعلن عن الثابت كما يعلن عن المتغير تماماً .. ولكن بإستخدام كلمة **Const** بدلاً من كلمة **Dim** .

• فمثلاً .. نعلن عن الثابت Tax والذى يمثل الضريبة .. كما بالصورة التالية :

```
Dim Salary As Integer
CONST TAX=100
```



يمكن أن تكتب حروف الثابت Capital ، أو Small أيضاً .

• **فائدة الثوابت** .. تقليل الوقت وتسهيل العمل فى البرنامج وتقليل الخطأ .

• الإعلان عن الثابت النصى :

يمكن الإعلان عن ثابت نصى يمثل أسم الشركة .. كما بالصورة التالية :

```
CONST Company_Name " Vodafone"
```



تم وضع الثابت النصى فى الكود السابق بين علامتى "" تنصيص بعكس الثابت الرقمى .

• الإعلان عن الثابت من النوع التاريخ :

يتم الإعلان عن ثابت ليعبر عن تاريخ السداد .. كما بالصورة التالية :

```
CONST Pay_Date=#01/01/2014#
```

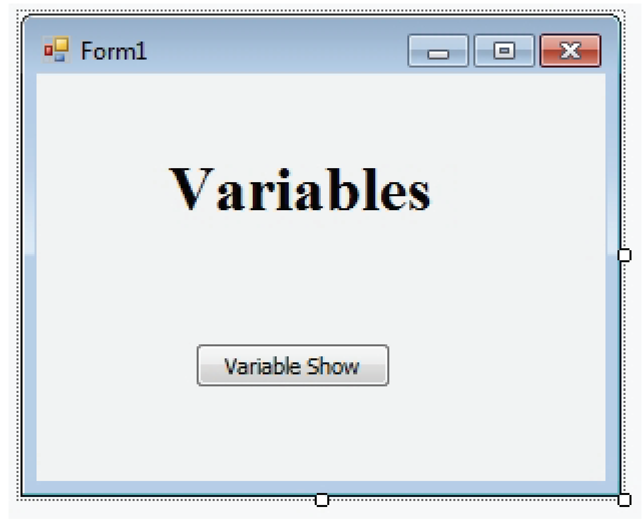
▪ مثال على المتغيرات والثوابت 1 :

نقوم فى هذا المثال بتعريف مجموعة من المتغيرات ، والتى تمثل مجموعة من أسماء الموظفين ، ووظيفة كل منهم .. ولتنفيذ ذلك إتبع الخطوات التالية :

1 – إفتح مشروع جديد New Project .

2 – أضف عنوان Label وغير قيمة الخاصية Text لتصبح Variable .

3 – أضف مفتاح Button وغير قيمة الخاصية Text لتصبح Show .. فيظهر النموذج .. كما بالشكل التالى :

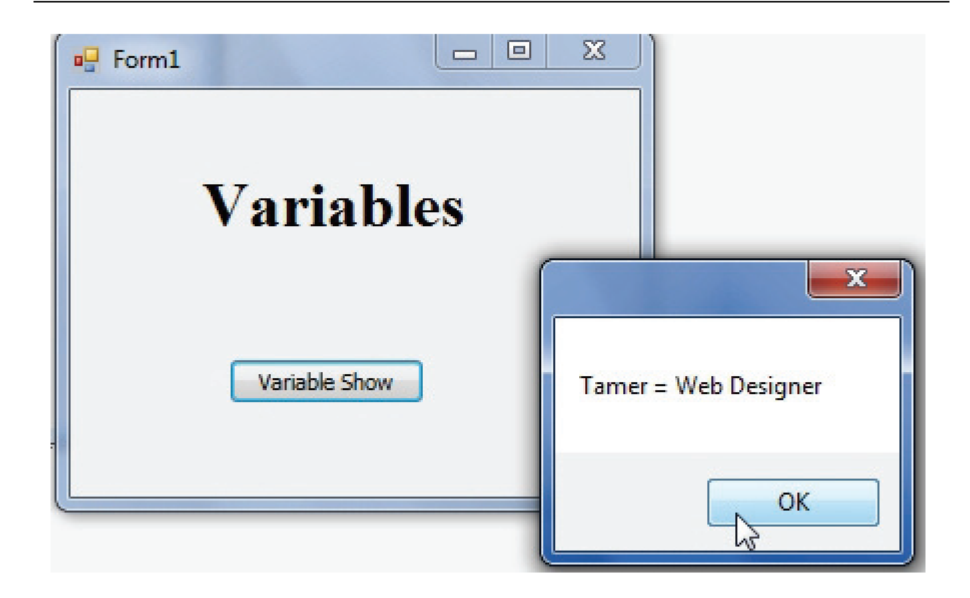


4 – إضغط مرتين على المفتاح Button .. فتظهر نافذة الكود .

5 – قم بتعريف كل من الأسم ، والوظيفة .. كما بالكود التالى :

```
Dim Tamer, Ali, Ahmed, Amr, Hany
Tamer = "Web Designer"
Ali = "Programmer"
Ahmed = "IT"
Amr = "Graphic Designer"
Hany = "System Analysis"
MessageBox.Show("Tamer = Web Designer")
MessageBox.Show("Ali = Programmer")
MessageBox.Show("Amr = Graphic Designerr")
MessageBox.Show("Hany = System Analysis")
```

6 – شغل البرنامج .. بالضغط على مفتاح **F5** .. ثم إضغط مفتاح **Show Variable** .. فتظهر رسالة بالمتغير الأول كما بالشكل التالى :



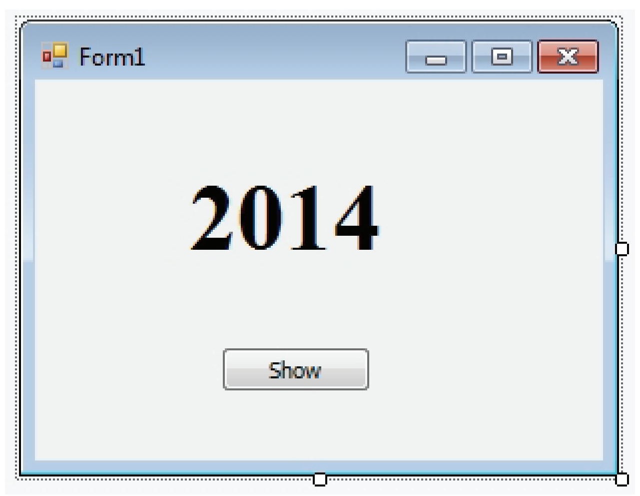
7 – كرر عملية الضغط على مفتاح **Ok** .. ليظهر أسم ووظيفة باقى الموظفين كما بالشكل التالى :



■ مثال 2 :

نقوم فى هذا المثال بتعريف متغير بدون تعريف نوع هذا المتغير .. ولتنفيذ ذلك إتبع الخطوات التالية :

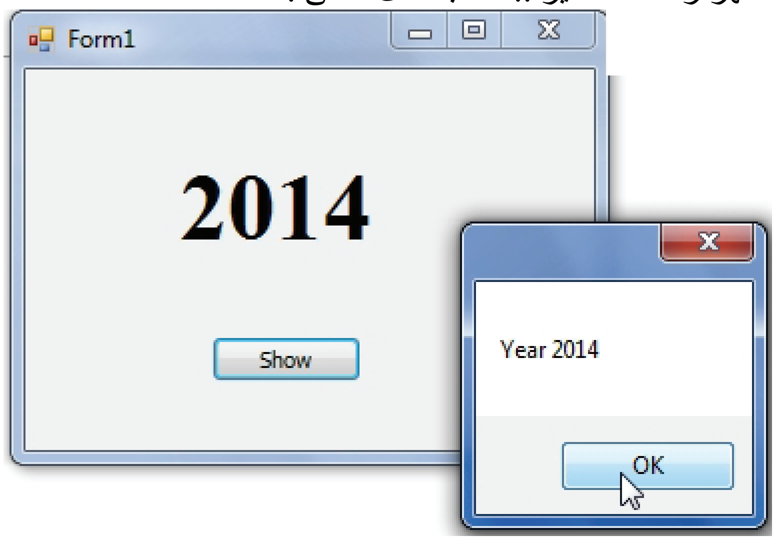
1 – إفتح مشروع جديد .. كما تعلمت سابقاً ، ثم أضف عنوان للنموذج ، ومفتاح للتحكم .. كما بالشكل التالى :



2 – إضغط مرتين على المفتاح **Show** .. فتظهر نافذة الكود .. ثم أكتب الكود التالى :

```
Dim Year = "2014"
MessageBox.Show("Year 2014")
```

3 – شغل البرنامج .. بالضغط على مفتاح **F5** .. ثم إضغط مفتاح **Show** فتظهر رسالة المتغير .. كما بالشكل التالى :



■ التعليقات Comments :

هى الملاحظات التى توضع على الكود المكتوب لشرحة أحياناً ، أو لعرض وظيفة الـ Procedure فى أحيان أخرى .. أو حتى تحمل بيانات شخصية للمبرمج نفسه .

- المهم أن هذه التعليقات لا يراها الـ Compiler ، ولا يقوم بتنفيذها ، ولا يقوم بتصحيحها داخل البرنامج .. فهى مجرد ملاحظات للعرض فقط .

- ولكتابة تعليق نستخدم كلمة **Rem** .. وهى إختصار للكلمة **Remark** .. ولتنفيذ كتابة التعليقات Comments أكتب الكود التالى داخل نافذة الكود :

```
Rem This Code to Explin Constsnts
Dim Salary As Integer
Const TAX = 100
Const Company_Name = "Vodafone"
Const Pay_Date = #1/1/2014#
```



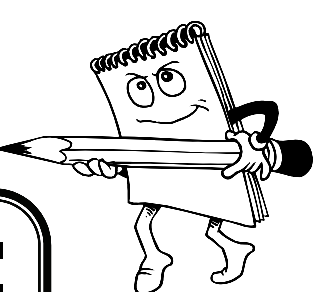
- بعد كتابة سطر الملاحظات ، والضغط على مفتاح **Enter** يأخذ الكود اللون الأخضر .. وهو لون مختلف عن باقى السطور .
- يمكن كتابة الجملة **Rem** فى نهاية سطر الكود .. وذلك بعد كتابة الكود نفسه وليس قبلة .

- يمكن إستخدام كلمة **Rem** فى تعطيل تنفيذ أى كود برمجى داخل لغة الـ Visual Basic .. وذلك بوضع كلمة **Rem** قبل بداية الكود .. كما بالشكل التالى :

```
Dim Year = "2014"
REM MessageBox.Show("Year 2014")
```

- شغل البرنامج بالضغط على مفتاح **F5** .. نجد أن الكود لا يعمل .

الفصل السابع
المصفوفات Arrays



نتعلم فى هذا الفصل ..

- تعريف المصفوفات Arrays .
- المصفوفات المتغيرة .
- المصفوفات متعددة الأبعاد .
- عيوب المصفوفات .
- أمثلة على أنواع المصفوفات المختلفة .

الفصل السابع

المصفوفات Arrays

بعد أن تعرفنا على الثوابت والمتغيرات فى الفصل السابق .. نريد الآن الإعلان عن متغير لإستقبال بيانات .. مثل بيانات درجات الطلاب فى أحد المواد الدراسية .. وذلك بكتابة الجملة :

```
Dim Mark As Byte
```

ويستطيع بعد ذلك المستخدم كتابة الدرجة بعد كلمة Mark ، أو حتى بعد تخصيص قيمة له داخل الكود .. وذلك بكتابة الجملة ..

```
Mark = 20
```

- فإذا أرادنا إدخال درجات 3 طلاب لنفس المادة الدراسية .. هنا نقوم بتعريف 3 متغيرات وهى (Mark 1 ، Mark 2 ، Mark 3) .

- لو إعتبرنا أن إدخال درجة كل طالب من خلال الجملة Mark أمر منطقى .. فماذا نفعل لو نريد إدخال بيانات مدرسة بالكامل مكونة من نريد إدخال بيانات مدرسة بالكامل مكونة من 300 طالب ؟ .. هل سنعلن عن 300 متغير ؟ فالأمر غير منطقى .

- **من هنا تأتى فكرة المصفوفة Array ..**

■ **المصفوفة Array :**

- هى وسيلة تستخدم لتخزين مجموعة من القيم ، أو البيانات التى لها نفس النوع .

- وللإعلان عن المصفوفات Array .. نستخدم الجملة **Dim** مثل المتغيرات تماماً ، ثم يليها أسم المصفوفة Array .

- **ولكن** .. يجب أن يأتى بعد أسم الـ Array عدد العناصر التى سوف يحتويها الـ Array ، ويوضع بين قوسين ( ).

- **فمثلاً** .. إذا أرادنا عمل مصفوفة Array للجملة Mark لتصبح :

```
Dim Mark ( 15 ) As Byte
```

- **وتعنى الجملة السابقة** .. أننا قمنا بعمل Array لعدد 16 عنصر ، لأن العد يبدأ من الرقم صفر .. أى من (0 – 15) .



الرقم بين القوسين يمثل الـ Index الذى إستخدمناها أثناء التعامل مع عناصر الـ Array .

- **فمثلاً** .. إذا أرادنا تخصيص قيمة للعنصر الأول منها .. تكون الجملة كما بالكود التالى :

```
Mark ( 0 ) = 19
```

- **وتعنى الجملة السابقة** .. أن أول عنصر فى الـ Array يحمل القيمة 19 .

- أما إذا أرادنا إعطاء قيمة للعنصر الثانى .. تكون الجملة كما بالكود التالى :

```
Mark ( 0 ) = 19
Mark ( 1 ) = 15
```

وهكذا لباقى العناصر .



كثير من المبرمجين يعانـون من التعامل مع الـ Array بإستخدام الـ Index 0 فهنا أتاح لنا الـ Visual Basic إمكانية تحديد بداية ، ونهاية الـ Index Array.

• **ولإيضاح ذلك :**

إذهب للمصفوفة Mark 15 .. وأكتب داخل القوسين **(1 To 16)** بدلاً من كتابة الرقم 15 .. كما بالكود التالى :

```
Dim Mark(1 To 16) As Byte
```

• **وتعنى الجملة السابقة** .. أن المصفوفة Mark تتكون من 16 عنصر ، يبدأ العنصر الأول من Index 1 ، والأخير Index 16 .



بعد كتابة الجملة السابقة (1 To 16) لابد من إلغاء الجملة Mark (0) لأنه لم يعد موجود القيمة صفر .. لذلك نقوم بتغييرها لتصبح Mark (1) ، والجملة الثانية تصبح Mark (2).

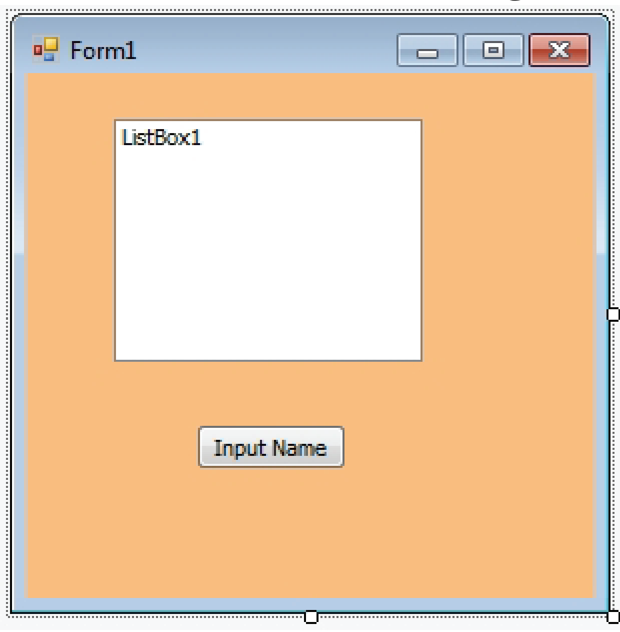
■ **المصفوفات المتغيرة Dynamic Array :**

وتقوم فكرة المصفوفة المتغيرة بأنه .. يمكن تعريف المصفوفة بدون تحديد عدد العناصر الموجودة بهذه المصفوفة .. مثل فكرة إدخال البيانات داخل جدول قاعدة البيانات .. وذلك من خلال صندوق حوار **Input Box** يسأل المستخدم عن إدخال البيان الجديد .



صمم برنامج .. يقوم المستخدم من خلاله بإدخال الأسم Name ، وذلك من خلال مربع حوار Input Box .. يسمح للمستخدم بإدخال أى عدد من الأسماء ثم تخزين هذه الأسماء داخل الذاكرة .. ولتنفيذ ذلك .. إتبع الخطوات التالية :

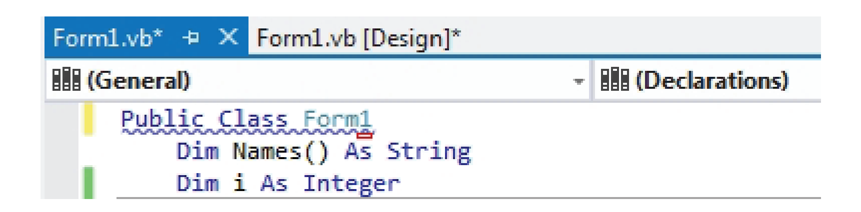
1 – إفتح مشروع جديد New Project .
2 – أضف List Box ، ومفتاح تحكم Buttons من صندوق الأدوات .
3 – غير الخاصية Text .. لتصبح Input Name .. فيظهر شكل النموذج كما بالشكل التالى :



4 – إضغط مرتين على نافذة النموذج Form 1 .. فتظهر نافذة النموذج .. ثم قم بكتابة الكود التالى فى الجزء الخاص بـ Public Class Form1 :

```
Dim Names( ) As String
Dim i As Integer
```

ليظهر الجزء الخاص بالكود من نافذة الأكواد .. كما بالشكل التالى :



5 – إضغط مرتين على المفتاح **Input Name** .. فتظهر نافذة الكود .. ثم أكتب الكود التالى :

```
Dim Name As String
    Name = InputBox("Enter New Name" & vbNewLine & "Enter e to Exit")
    While Name <> "e"
      ReDim Preserve Names(i)
      Names(i) = Name
      i = i + 1
      Name= InputBox("Enter New Name" & vbNewLine & "Enter e to Exit")
 End While
```

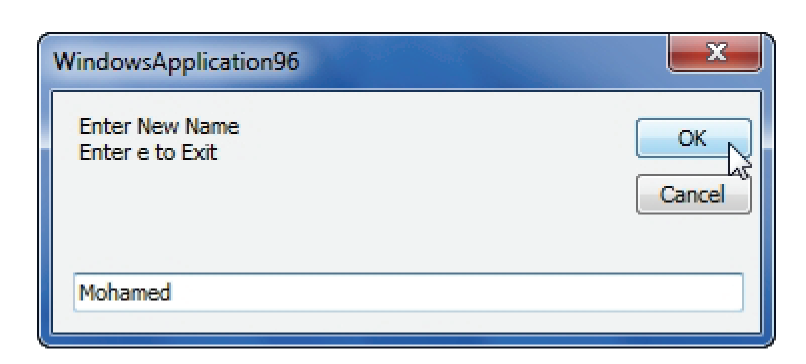
معنى الكود السابق !!

- تم تعريف المتغير Name من النوع String ويأخذ القيمة I .. وذلك فى الجزء الخاص بـ Public Class Form1 .
- الكود الخاص بمربع الحوار الذى سيسألنا عن إدخال أسم جديد من خلال الدالة **Input Box** .. حيث تم وضع العلامة & داخل الكود لظهور مربع الحوار مرة أخرى لإدخال الأسم التالى .. و هكذا .
- الكود الخاص بمربع الحوار الذى سيسألنا عن إدخال أسم جديد من خلال الدالة **Input Box** .. حيث تم وضع العلامة & داخل الكود لظهور مربع الحوار مرة أخرى لإدخال الأسم التالى .. و هكذا .
- الجملة **ReDim** مع الجملة Preserve تستخدم لتغيير عناصر المصفوفة بدون تغيير البيانات المخزنة بها .

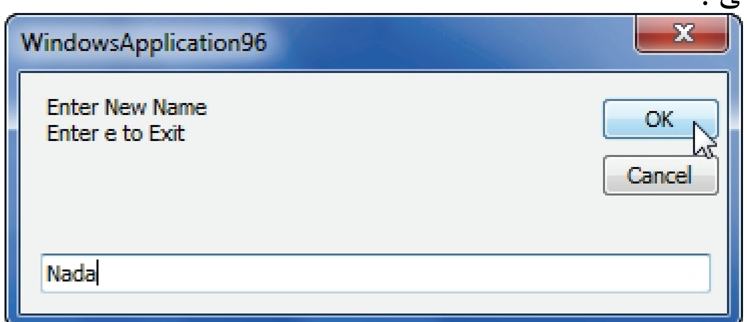
- الجملة **While** .. لإظهار مربع الحوار أكثر من مرة .. وذلك لإدخال بيان جديد.

- الحرف **e** فى الجملة While Name .. وذلك لإنهاء إدخال البيانات فى مربع الحوار .. وذلك بمجرد كتابة الحرف **e** .. ثم يقوم البرنامج بعدها بتنفيذ الجملة End While .

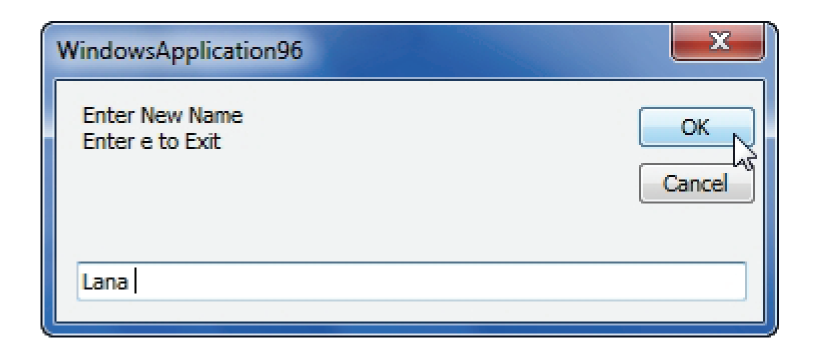
6 – شغل البرنامج .. بالضغط على مفتاح **F5** .. ثم إضغط مفتاح **Input Name** .. فيظهر مربع الحوار لإدخال الأسم الأول ..كما بالشكل التالى :



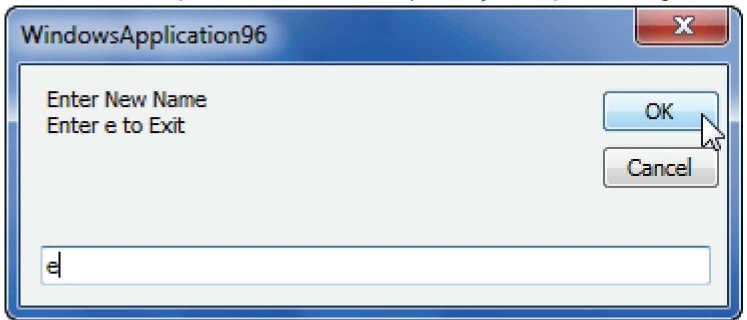
7 – إدخل الأسم الأول .. ثم إضغط **OK** ، ثم إدخل الأسم الثانى .. كما بالشكل التالى :



8 – إدخل الأسم الثالث .. ثم إضغط **OK** .. و هكذا لباقى أسماء المصفوفة ..



9 – أكتب حرف **e** .. وذلك لإنهاء إدخال بيانات المصفوفة ..





لم تظهر البيانات التى تم إدخالها فى مربع الحوار داخل الـ List Box فى النموذج Form .. ولكنها مخزنة داخل الذاكرة .. ويمكن إظهار هذه البيانات عن طريق إضافة مفتاح أخر للنموذج لإظهار البيانات .

■ **المصفوفات متعددة الأبعاد Multi Dimensional Arrays :**

فى بعض الحيان نريد حفظ بيانات أكثر من متغير فى نفس المصفوفة .. مثل تسجيل أسماء ودرجات الطلاب ، أو أسماء المدن ودرجات الحرارة الخاصة بها ففى هذه الحالة نستخدم مصفوفة متعددة الأبعاد لتخزين مثل هذه البيانات .

• **فمثلاً** .. الجدول التالى يمثل درجات وأسماء الطلاب فى 3 مواد دراسية :

2	1	0		
العلوم	الرياضيات	الحاسب الآلى	أسم الطالب	
40	50	45	وليد وحيد	0
46	47	44	خالد عادل	1
38	32	35	أحمد جلال	2
42	48	33	ياسمين طارق	3

حيث يتم تمثيل كل درجة من هذه الدرجات عن طريق رقم الصف ، ورقم العمود الذى تقع به هذه الدرجة .. فمثلاً العمود الأول يأخذ الرقم 0 ، والعمود الثانى يأخذ الرقم 1 ، والعمود الثالث يأخذ الرقم 2 .. وهكذا .

- ويتم تمثيل بيانات الطالب أحمد جلال بالقيمة (2 , 1) .. وهى تعنى الصف الثالث مع العمود الثانى .

- ولتعريف البيانات السابقة .. نستخدم **مصفوفة ثنائية الأبعاد** .. وهى نفس فكرة إدخال بيانات المصفوفة السابقة .. ولكن مع تحديد عدد الصفوف والأعمدة بها وتأخذ المصفوفة الكود التالى :

```
Dim Degrees ( 1 , 2 ) As Single
Degree ( 0,0) = 45
Degree ( 0,1) = 50
Degree ( 0,2) = 40
Degree ( 1,0) = 44
```



صمم برنامج يمكن من خلاله إدخال أسم الطالب ، والدرجة وذلك بالضغط على مفتاح التحكم **Add** .. ولتنفيذ ذلك .. إتبع الخطوات التالية :

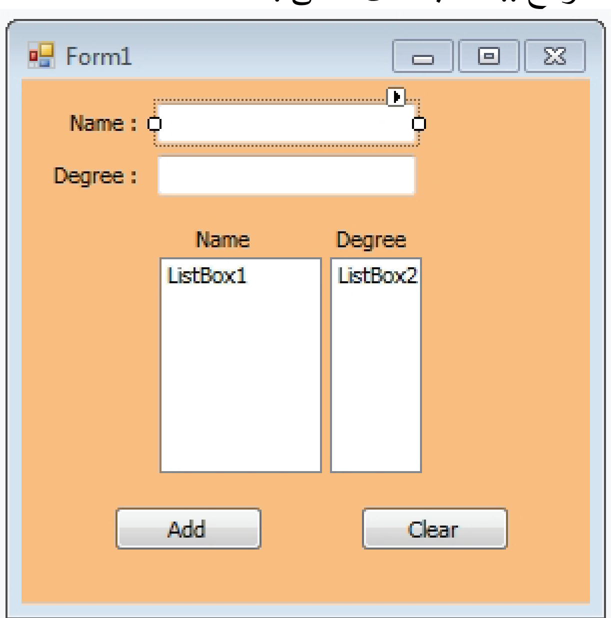
1 – إفتح مشروع جديد New Project .

2 – أضف عدد 2 Lebel .. وغير الخاصية Text للعنصر الأول لتصبح Name ، والثانى لتصبح Degree .

3 – أضف عدد 2 List Box .. ليظهر أسم الطالب فى المربع الأول ، والدرجة فى المربع الثانى .

4 – أضف عدد 2 مفتاح تحكم Button .. وغير الخاصية Text للمفتاح الأول لتصبح **Add** ، والمفتاح الثانى لتصبح **Clear** .

فيظهر شكل النموذج .. كما بالشكل التالى :



5 – إضغط مرتين على المفتاح **Add** .. فتظهر نافذة الكود .. ثم أكتب الكود التالى :

```
If TextBox1.Text = "" Then
    MessageBox.Show("Enter your Name")
  Else
    ListBox1.Items.Add(TextBox1.Text)
  End If
  If TextBox2.Text = "" Then
    MessageBox.Show("Enter your Degree")
  Else
    ListBox2.Items.Add(TextBox2.Text)
End If
```



- **أنه فى حالة عدم إدخال أى بيانات فى الـ Text Box1 الخاص بالأسم Name .. فعند الضغط على المفتاح Add تظهر رسالة للمستخدم تطلب منه إدخـال الأسـم Name .. وإلا سيتم إضـافة الأسـم الذى سيتم إدخـالـة فى الـ List Box 1 .**

- **أنه فى حالة عدم إدخال أى بيانات فى الـ Text Box 2 الخاص بالدرجة Degree .. فعند الضغط على المفتاح Add تظهر رسالة للمستخدم تطلب منه إدخال الدرجة Degree .. وإلا سيتم إضافة الدرجة الى سيتم إدخالها فى الـ List Box 2 .**

6 – إضغط مرتين على المفتاح **Clear** .. وأكتب الكود التالى :

```
ListBox1.Items.Clear
ListBox2.Items.Clear
```



- **أنه عندما يتم تحديد الأسم والدرجة من القائمة List Box .. بمجرد الضغط على المفتاح Clear .. يتم حذف البيانات التى تم إدخالها .**

7 – إضغط مرتين على الـ Text Box 1 .. فتظهر نافذة الكود .. ثم إختار الحدث Click من القائمة المنسدلة Procedure .. ثم أكتب الكود التالى :

```
TextBox1.Text=""
```

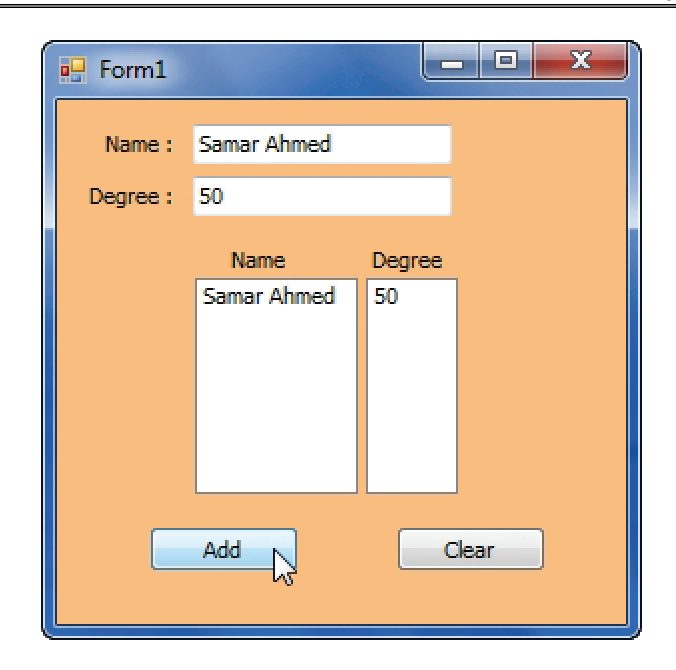
8 – إضغط مرتين على الـ Text Box 2 .. فتظهر نافذة الكود .. ثم إختار الحدث Click من القائمة المنسدلة Procedure .. ثم أكتب الكود التالى :

```
TextBox2.Text=""
```



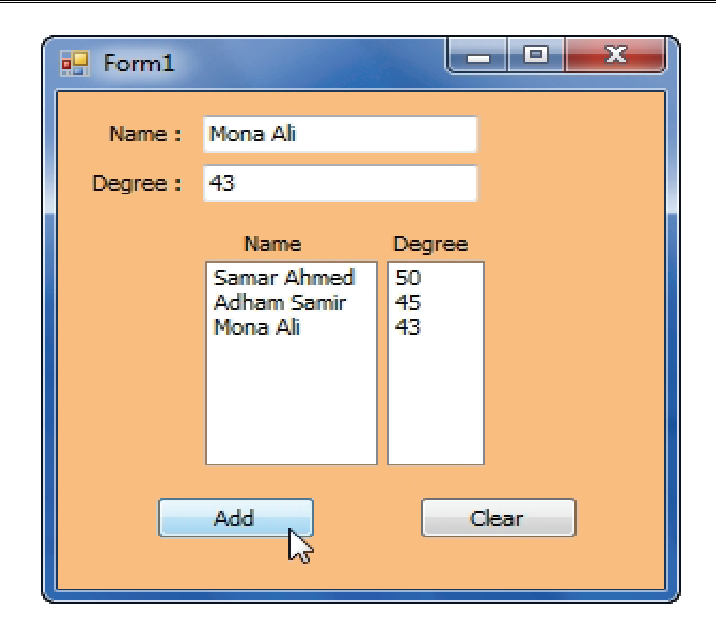
- **أنه بمجرد وضع مؤشر الـ Mouse داخل مربع الأسم Name ، أو الدرجة Degree لإدخال بيانات جديدة .. يتم تفريغ مربع الإدخال تلقائياً من البيانات .**

9 – شغل البرنامج .. بالضغط على مفتاح **F5** .. ثم إدخل بيانات .. كما بالشكل التالى :



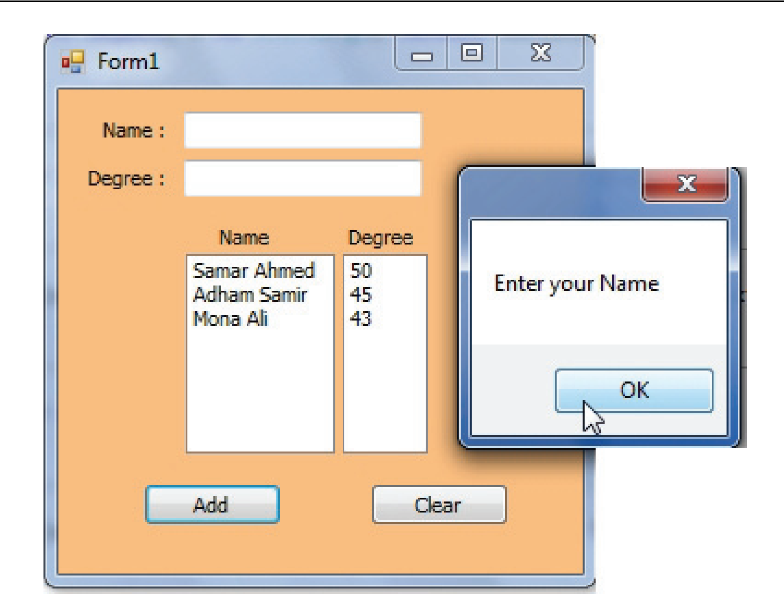
- بعد إدخال أسم ودرجة الطالب الأول .. يتم الضغط على مفتاح **Add** لتتم الإضافة .. ولإدخال أسم ودرجة الطالب الثانى .. بمجرد الوقوف بمؤشر الـ Mouse داخل مربع الإدخال .. يتم تفريغ الحقول من البيانات .

- تظهر بيانات الطلاب التى تم إدخالها .. كما بالشكل التالى :

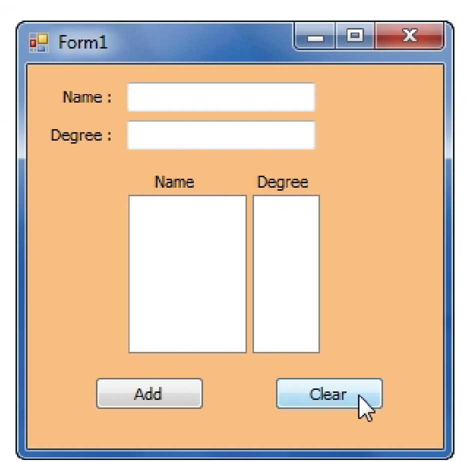


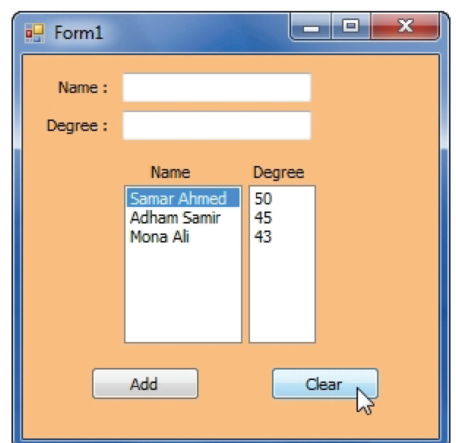


فى حالة عدم إدخال أى بيانات .. أى حقول الإدخال فارغة .. فعند الضغط على المفتاح Add فتظهر رسالة التنبية .. كما بالشكل التالى :



• ولحذف البيانات من القائمة .. حدد البيان .. ثم إضغط مفتاح **Clear** فتتم عملية الحذف .. كما بالشكل التالى :







• تم حفظ بيانات النموذج السابق داخل الذاكرة المؤقتة **Ram** .

- **ولحفظ البيانات بشكل دائم ، وبكميات ضخمة .. يتم ذلك من خلال ربط نماذج الـ Visual Basic بقواعد البيانات Access ، أو SQL Server وهو ما سنتعلمة فى الفصل الخاص بقواعد البيانات Data Base .**

■ عيوب المصفوفات :

1 – أنها بناء غير ديناميكى Not Dynamic Structure :

بمعنى أن عملية تغيير أبعاد المصفوفة .. أى إزاحة القيم بعد حذف قيمة معينة داخل المصفوفة .. تستغرق الكثير من الوقت .

2 – أنه يتم التعامل مع عناصر المصفوفة عن طريق موضع العنصر Index داخل المصفوفة فقط .. والذى لا يدل على محتوى هذا الموضع .. حيث يمكن وضع العنصر فى أى مكان داخل المصفوفة .



تتغلب المصفوفة الديناميكة Dynamic Array على عيوب المصفوفات .. بحيث يتم تحريك العناصر تلقائياً للموضع الجديد .

الفصل الثامن
الجمل الشرطية والحلقات التكرارية

نتعلم فى هذا الفصل ..



■ **أولاً : الجمل الشرطية :**

- الجملة IF .. Then .
- الجملة IF .. Then .. Else .
- الجملة Select Case .

■ **ثانياً : الحلقات التكرارية :**

- الحلقة For .. Next .
- الحلقة Do .. While .
- الحلقة While .. End While .

الفصل الثامن
الجمل الشرطية والحلقات التكرارية

نحتاج فى كثير من الأحيان إختبار شروط معينة للتأكد من صحة ، أو عدم صحة هذه الشروط .. مثل إختبار أسم المستخدم ، وكلمة السر عند بداية تشغيل البرنامج .. فإذا كانت البيانات صحيحة بالفعل .. سيتم إدخال المستخدم الحالى للبرنامج ، أما إذا كانت غير صحيحة يتم الخروج من البرنامج .

وتحتوى لغة الـ Visual Basic على عدة جمل شرطية .. وهى :

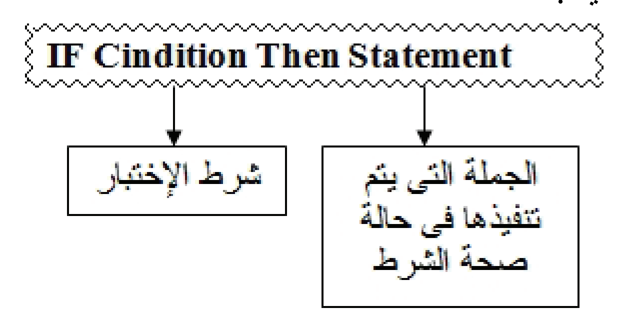
1 – IF .. Then
2 – IF .. Then .. Else
3 – Select Case

■ أولاً : الجمل الشرطية :

1 – الجملة IF .. Then :

تستخدم هذه الجملة لإختبار شرط معين .. وفى حالة صحة هذا الشرط يتم تنفيذ الجملة ، أو مجموعة الجمل التى تليها .. ويتم تنفيذ هذه الجملة بإحدى صيغتين :

• **الصيغة الأولى : صيغة السطر الواحد Single Line :**
وتكون كما بالصيغة التالية :



• **فمثلاً ..**

IF X >10 Then X=0



يمكن تنفيذ أكثر من جملة لـ IF .. Then فى وقت واحد ، وذلك بوضع علامة : Colon بين كل جملة ، والجملة التى تليها .. وتكون كما بالصيغة التالية :

```
IF Condition Then Statement : Statement : Statement
```

- **فمثلاً ..**

```
IF X >10 Then X=0 : Y=1
```

- **الصيغة الثانية : صيغة الأسطر المتعددة Multi Line :**

وتكون كما بالصيغة التالية :

```
IF Condition Then
Statement 1
Statement 2
..............
..............
Statement N
End IF
```

- يتضح لنا من الصيغة السابقة .. أن الجملة التى يتم تنفيذها توجد فى السطر التالى للجملة **IF** .. وليس فى نفس السطر .. وبالتالى فلابد من وجود جملة ، أو كلمة تحدد نهاية الجملة **IF** .. وهى الجملة **End IF** والتى تخبر المترجم Compiler الإنتهاء عند هذا الموضع .

- إذا كان الشرط صحيح .. يتم تنفيذ جميع الجمل الموجودة بين الجملة **IF** ، والجملة **End IF** .

• وإذا كان الشرط غير صحيح .. يتم تنفيذ أول جملة بعد جملة **End IF** داخل البرنامج .

• **فمثلاً ..**

```
IF X > 10  Then
    X = 0
    Y = 1
End IF
```

2 – الجملة IF .. Then .. Else :

• فى هذه الجملة يتم إختبار جملة ، أو مجموعة جمل .. فى حالة تحقق الشرط الخاص بالجملة **IF** .

• أما فى حالة عدم تحقق الشرط الخاص بالجملة **IF** .. سوف يتم تنفيذ جملة ، أو مجموعة جمل أخرى .

• وتكون الصيغة العامة للجملة **IF .. Then .. Else** .. كما بالصيغة التالية :

```
IF Condition Then
        Statement Block 1
Else
        Statement Block 2
End IF
```

• هنا تعمل الجملة **IF** على إختبار الشرط .. فإذا تحقق هذا الشرط .. سيتم تنفيذ الجملة ، أو مجموعة الجمل الموجودة أسفل الجملة **IF** .

• أما إذا لم يتحقق الشرط .. سيتم تنفيذ الجملة ، أو مجموعة الجمل الموجودة أسفل الجملة **Else** .

• **فمثلاً ..** لو الشرط صحيح يتم تنفيذ الجملة التى تلى جملة **IF** وتجاهل الجملة **Else** ، والعكس .. كما فى الكود التالى :

```
IF X < 10  Then
    X = 0
Else
    Y = 1
End IF
```

• لكن فى بعض الأحيان نحتاج لتنفيذ عمليات مختلفة ، وقيم مختلفة لمتغير معين فيتم ذلك عن طريق إستخدام الجملة **IF .. Then .. Else** .. كما بالصيغة التالية :

```
IF Condition 1 Then
    Statement Block 1
Else IF Condition 2 Then
     Statement Block 2
Else IF Condition 3 Then
     Statement Block 3
Else
      Statement Block 4
End IF
```



نتعلم فى هذا المثال إستخدام الجملة الشرطية **IF .. Else** داخل الجملة **IF .. Then .. Else :**

صمم برنامج يقوم بإظهار تقدير الطالب عندما يتم إدخال قيمة الدرجة فى مربع الإدخال Score ، وعند الضغط على المفتاح Dgree يتم عرض رسالة بتقدير هذا الطالب .. حيث أن التقديرات تكون كما يلى :

• لو الدرجة أكبر من أو تساوى **90** .. يكون التقدير (**ممتاز**) .

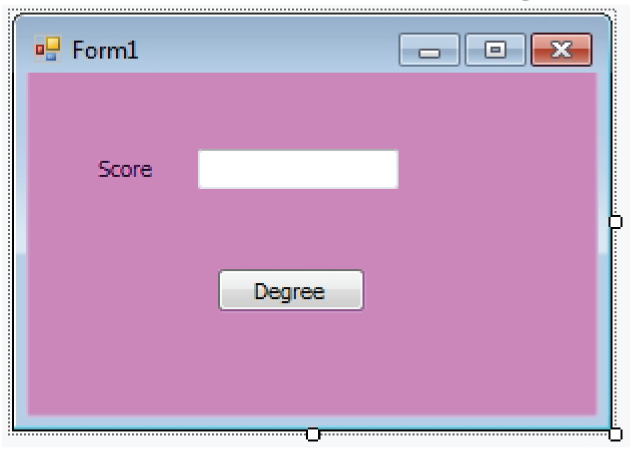
- لو الدرجة أكبر من أو تساوى **80** .. يكون التقدير (**جيد جداً**) .
- لو الدرجة أكبر من أو تساوى **70** .. يكون التقدير (**جيد**) .
- لو الدرجة أكبر من أو تساوى **50** .. يكون التقدير (**مقبول**) .
- لو الدرجة أكبر من أو تساوى **35** .. يكون التقدير (**ضعيف**) .
- لو الدرجة أقل من **35** .. يكون التقدير (**ضعيف جداً**) .

ولتنفيذ ذلك إتبع الخطوات التالية :

1 – إفتح مشروع جديد New Prohect .

2 أضف Text Box لإدخال الدرجة ، ومفتاح للتحكم Button ، وعنوان Label .

3 – غير الخاصية Text للمفتاح Button 1 لتصبح Degree ، ثم غير الخاصية Text للعنوان Label لتصبح Score .. فتظهر نافذة النموذج كما بالشكل التالى :



4 – إضغط مـرتين على المفتاح Degree .. فتظهر نافذة الكود ، ثم أكتب الكود التالى :

```
Dim X  As  Single
X = TextBox1.Text
If X >= 90 Then
    MsgBox  ("ممتاز")
ElseIf X >= 80 And  X < 90 Then
   MsgBox ("جيد جداً")
ElseIf X >= 70 And  X < 80 Then
    MsgBox ("جيد")
ElseIf X >= 50 And  X < 70 Then
    MsgBox ("مقبول")
ElseIf X >= 35 And  X < 50 Then
   MsgBox ("ضعيف")
Else
   MsgBox ("ضعيف جداً")
```

- **تم تعريف متغير X من النوع Single ليعبر عن الدرجة .**
- **إستخدام الجملة IF لتعريف الشرط الأول ، ثم الجملة Then فى نهاية كل من السطر الأول إلى السطر الخامس .. للإنتقال لكتابة الشرط التالى .**
- **كتابة الرسالة الخاصة بتقدير كل شرط بعد الجملة Then .**
- **إستخدام الجملة Else بمفردها في الشرط الأخير فقط ، ثم كتابة رسالة التقدير الخاصة بهذا الشرط .**

5 – شغل البرنامج .. بالضغط على مفتاح **F5** .. ثم قم بإدخال الدرجات فتظهر التقديرات .. كما بالشكل التالي .



3 – **الجملة** Select Case :

إذا أرادنا كتابة جملة شرطية لإختبار أكثر من قيمة لمتغير معين ، وإجراء أفعال مختلفة لهذه القيم .. أو إذا أرادنا إختبار 5 جمل مثلاً نحتاج كتابة الجملة IF Then وبداخلها 4 جمل Else IF Then .

- **إذن ..**

تكون كمية الأكواد كثيرة .. لذلك نستخدم الجملة الشرطية Select Case .

- حيث تقدم الجملة Select Case طريقة سهلة وبسيطة لإنجاز العمليات التى نحددها وفقاً للقيم المختلفة .

- وتكون الصيغة العامة للجملة Select Case كما بالصيغة التالية :

```
Select Case Test Value
   Case Value 1
      Statement Group 1
   Case 2
      Statement Group 2
End Select
```

■ **والصيغة السابقة تعنى :**

- الـقيمة Value ممكن أن تـكون متغـير Variable ، أو تعـبير حسابى Numeric Expression ، أو تعبير نصـى String .

- كل الجمل تبدأ بالكلمة Case .. ثم يليها التعبير أو القيمة التى يتم مقارنة بقيمة العنصر الموجود فى جملة الإختبار .

- وفى حالة تساوى القيمة ، أو التعبير الموجود فى الجملة Case .. فأنه يتم تنفيذ الجملة ، أو مجموعة الجمل الموجودة .. مثل Group 1 مع تجاهل باقى قيم الجمل التالية .

- أما إذا كانت القيمة Case لا تساوى قيمة العنصر .. فسوف يتم الإنتقال إلى الجملة Case التى تلى هذه الجملة لإختبارها مع قيمة العنصر Test Value .

- وإذا تساوت هذه القيمة مع القيمة Test Value .. يتم تنفيذ الجملة ، أو مجموعة الجمل الخاصة بالـ Case الحالية .. وتجاهل باقة الجملة Case الموجودة داخل الجملة الشرطية .

- أما فى حـالة عدم تساوى القيمة مع Case Value .. يتم الإنتقال إلى الجملة التى تليها .



- **الجملة الشرطية Select Case تنتهى بالجملة End Select .**

- **الجمل Case الموجودة داخل الجملة الشرطية Select Case .. يمكن أن تحتوى على قيمة واحدة فقط ، أو على مدى معين من القيم ، أو على مقارنة بقيمة معينة .**

ويمكن توضيح أداء الجملة Select Case .. من خلال المثالين التاليين :

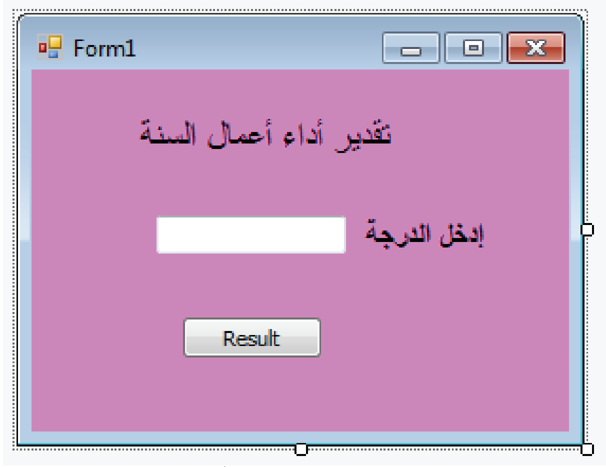


صمم برنامج لإظهار أداء أعمال السنة للتطبيقات العملية لمادة الحاسب الآلى لطلبة قسم نظم المعلومات .. بحيث تظهر التقديرات عندما تكون درجات أعمال السنة .. كما يلى :

- لو الدرجة من **10** إلى **20** .. يكون الأداء (**سيئ للغاية**) .
- لو الدرجة من **21** إلى **50** .. يكون الأداء (**سيئ**) .
- لو الدرجة من **51** إلى **60** .. يكون الأداء (**متوسط**) .
- لو الدرجة من **61** إلى **80** .. يكون الأداء (**جيد**) .
- لو الدرجة من**81** إلى **100** .. يكون الأداء (**ممتاز**) .

ولتنفيذ ذلك .. إتبع الخطوات التالية :

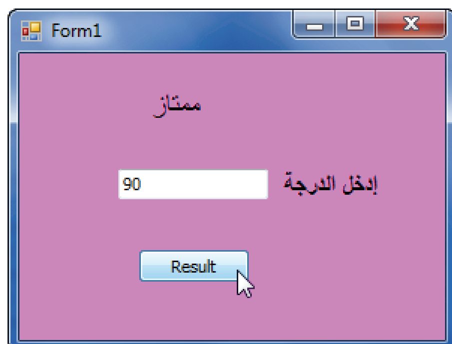
1 – إفتح مشروع جديد New Prohect .
2 – أضف Text Box لإدخال الدرجة ، ومفتاح للتحكم Button ، وعنوان Label
3 – غير الخاصية Text للمفتاح Button 1 .. لتصبح Result ، والخاصية Text للعنوان Label .. لتصبح (تقدير أداء أعمال السنة) .. فيظهر النموذج Form .. كما بالشكل التالى :

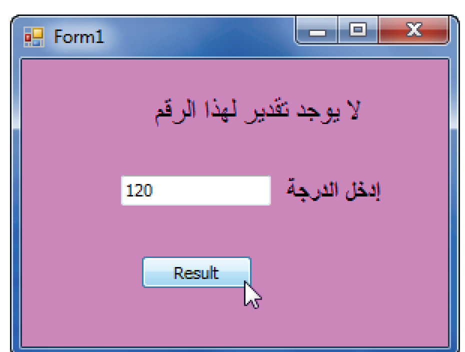


4 – إضغط مرتين على المفتاح **Result** .. وأكتب الكود التالى :

```
Dim X As Integer
  X = TextBox1.Text
  Select Case X
    Case 10 To 20
      Label1.Text = ("سيئ للغاية")
    Case 21 To 50
      Label1.Text = ("سيئ")
    Case 51 To 60
      Label1.Text = ("متوسط")
    Case 61 To 80
      Label1.Text = ("جيد")
    Case 81 To 100
      Label1.Text = ("ممتاز")
    Case Else
      Label1.Text = ("لا يوجد تقدير لهذا الرقم")
  End Select
```

5 – شغل البرنامج .. بالضغط على مفتاح **F5** .. ثم إدخل الدرجات فتظهر تقديرات الأداء .. كما بالشكل التالى :





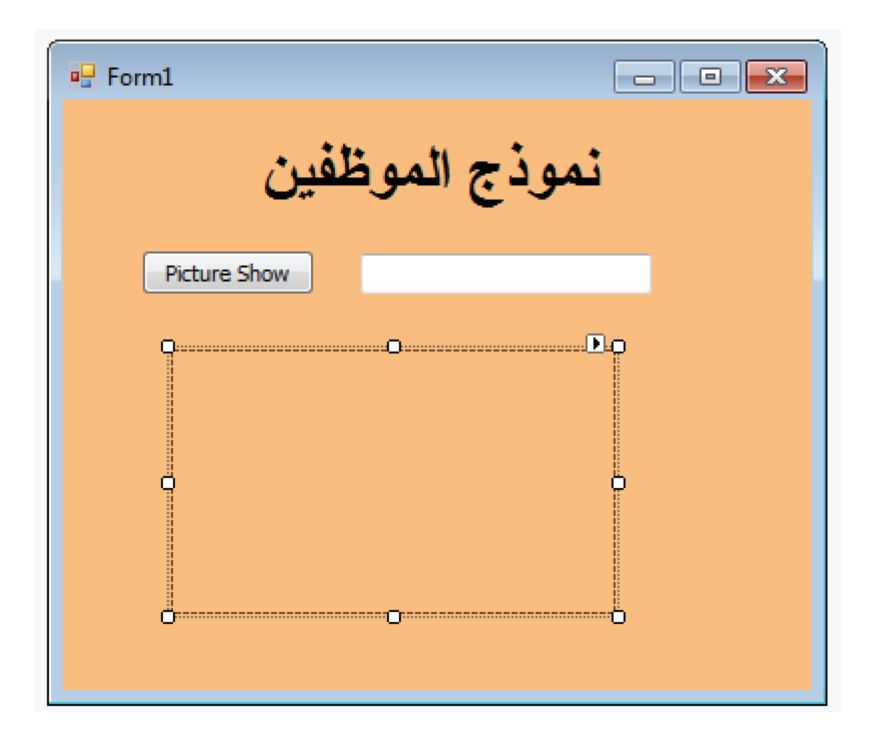


صمم برنامج لإظهار صورة الموظف بمجرد إدخال إسمه فى مربع الإدخال ثم الضغط على مفتاح عرض الصور .. ولتنفيذ ذلك .. إتبع الخطوات التالية :

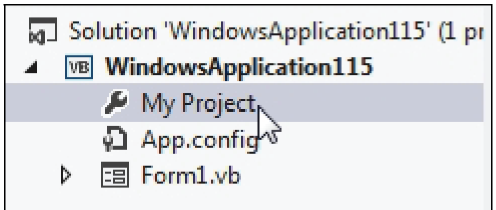
1 – إفتح مشروع جديد New Project .

2 – أضف Text Box لإدخال أسم الموظف ، مفتاح للتحكم Button ، وعنوان Label ، و Picture Box لنافذة النموذج .

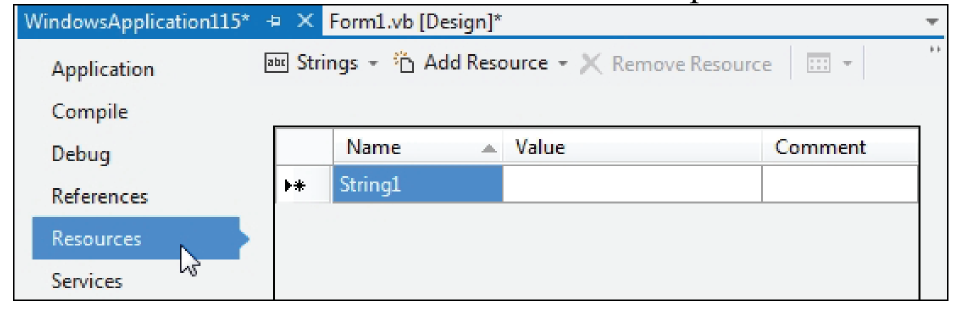
3 – غير الخاصية Text للمفتاح Button 1 .. لتصبح Show Picture ، والخاصية Text للعنوان Label 1 .. لتصبح (نموذج الموظفين) .. فتظهر نافذة النموذج كما بالشكل التالى :



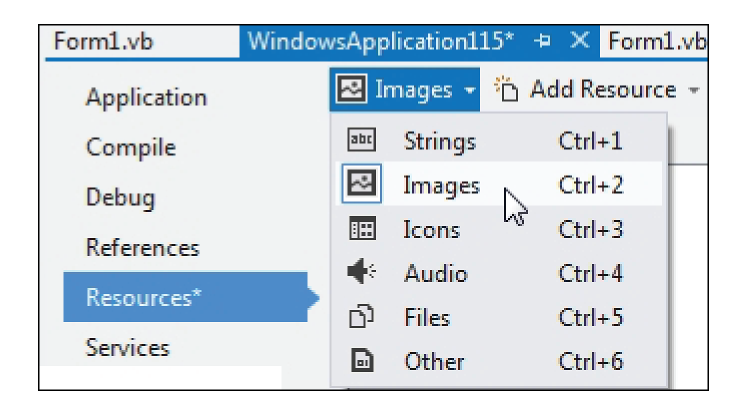
4 – إضغط مرتين الأسم My Project الموجود أعلى نافذة الخصائص .. كما بالشكل التالى :



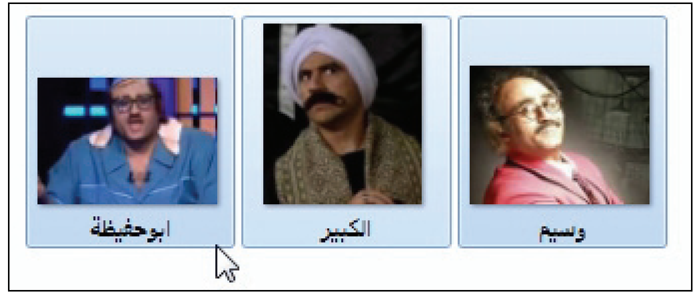
فتظهر النافذة Properties التالية :



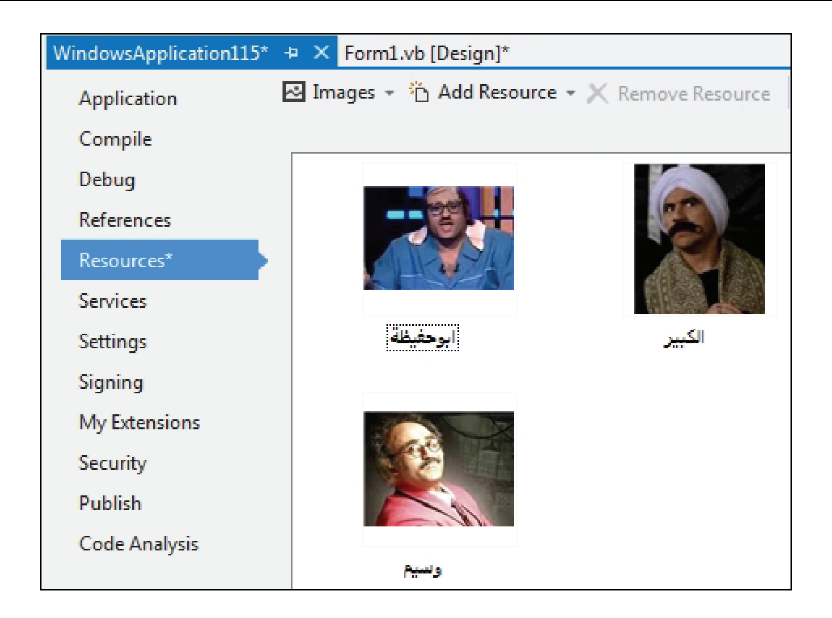
5 – إضغط على الأمر Resource .. ثم إضغط على القائمة المنسدلة Image وإختار منها الأمر Image .. كما بالشكل التالى :



6 – إذهب إلى المجلد Folder الموجود به الصور المطلوب إظهارها الموجود على جهاز الكمبيوتر .. كما بالشكل التالى :



7 – إسحب الصور بالـ Mouse .. ثم الإلقاء فى النافذة Image Properties كما بالشكل التالى :



8 – إضغط مرتين على المفتاح Picture Show من النافذة Form .. وأكتب الكود التالى :

```
Dim Name As String
  Name = TextBox1.Text
  Select Case Name
    Case "ابوحفيظة"
      PictureBox1.Image = My.Resources.ابوحفيظة
    Case "الكبير"
      PictureBox1.Image = My.Resources.الكبير
    Case "وسيم"
      PictureBox1.Image = My.Resources.وسيم
    Case Else
      MessageBox.Show("الأسم بهذا صورة لاتوجد")
 End Select
```

9 – شغل البرنامج .. بالضغط على مفتاح **F5** .. ثم إدخل الأسم الأول ، ثم الأسم الثانى .. وهكذا ، وإضغط على المفتاح **Picture Show** فتظهر الصور كما بالشكل التالى :







■ ثانياً : الحلقات التكرارية Loops :

تسمح الحلقات التكرارية بتنفيذ سطر معين ، أو عدة سطور من الكود بشكل متكرر .. حيث نحتاج فى كثير من الأحيان إلى تنفيذ بعض العمليات عدة مرات لذلك .. فإن الحلقات التكرارية Loops جزء هام فى أى لغة من لغات البرمجة .

وتدعم لغة الـ Visual Basic العديد من الحلقات التكرارية .. منها :

1 – الحلقة For .. Next .

2 – الحلقة Do .. Loop .

3 – الحلقة While .. End While .

1 – الحلقة التكرارية For .. Next :

تستخدم لمعرفة عدد المرات التى نريد تنفيذ فعل معين فيها .. وفى هذا النوع من الحلقات التكرارية يتم إستخدام متغير عددى كعداد Counter ، الذى يتم زيادته ، أو تقليل قيمتة داخل الجملة For .. Next ، والصيغة العامة للحلقة التكرارية For .. Next ، كما بالصيغة التالية :

```
For Count = Start to End [ Step Increment ]
        Statements
Next [ Counter ]
```

■ والصيغة السابقة للحلقة التكرارية For .. Next تعنى :

• العداد **Counter** .. يستخدم للعد داخل الحلقة التكرارية ، والقيمة **Start** هى القيمة الإبتدائية لهذا المتغير ، والقيمة **End** هى القيمة النهائية للمتغير Counter .

• **Step Increment** .. هى الخطوة ، أو القيمة التى يتم زيادة المتغير بها فى كل مرة يتم يتنفيذ الحلقة التكرارية .



المعطى Step Increment <u>إختيارى</u> أى ليس من الضرورى تحديده داخل الجملة For .. وفى حالة عدم تحديد هذا المعطى فأنه يتم زيادة الـ Counter بمقدار <u>واحد</u> فى كل مرة يتم تنفيذ الحلقة التكرارية بها .

• الجملة Next .. تستخدم لتحديد الموضع الذى تنتهى عنده الحلقة التكرارية .

• **يتم تنفيذ السطر ، أو مجموعة السطور التى ما بين الجملة For ، والجملة Next عدد من المرات حتى تصبح قيمة المتغير Counter أكبر من القيمة النهائية End .. وبعدها يتم الخروج من الحلقة التكرارية .**

■ ولإيضاح الصيغة السابقة للحلقة التكرارية For .. Next :

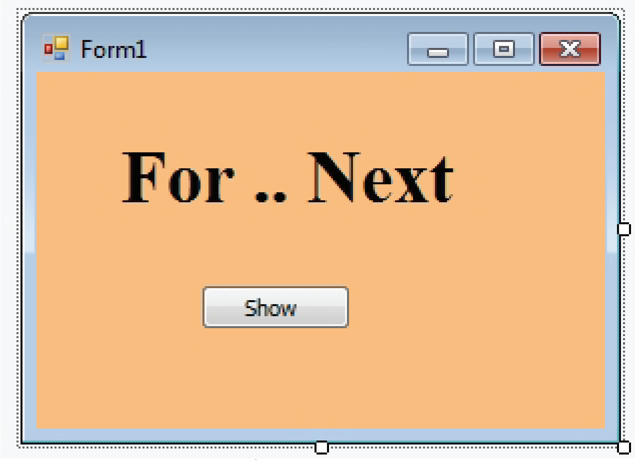
1 – يتم وضع قيمة المتغير Counter فى القيمة الإبتدائية Start .
2 – إختبار قيمة المتغير Counter .. فإذا كانت قيمتة أكبر من القيمة النهائية End يتم الخروج من الحلقة التكرارية .
3 – تنفيذ السطر أو مجموعة السطور الموجودة بين الجملة For ، والجملة Next .
4 – زيادة قيمة المتغير Counter بالقيمة المحددة فى المعطى Step Increment .
5 – تكرار الجمل السابقة .



صمم برنامج يقوم بإظهار الأرقام بداية من الرقم 1 إلى الرقم 10 .
ولتنفيذ ذلك إتبع الخطوات التالية :
1 – إفتح مشروع جديد New Project .

2 – أضف مفتاح تحكم Button ، وعنوان Label على النموذج Form .

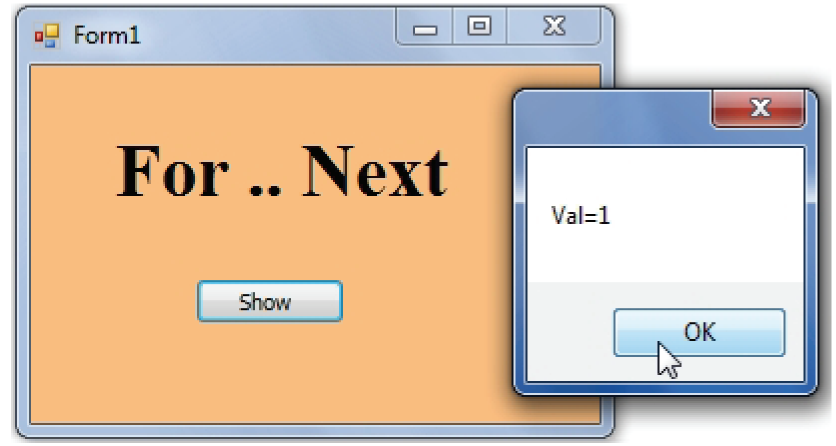
3 – غير الخاصية Text للمفتاح Button .. لتصبح Show ، والخاصية Text للعنوان Label لتصبح For .. Next ، فتظهر نافذة النموذج كما بالشكل التالى :



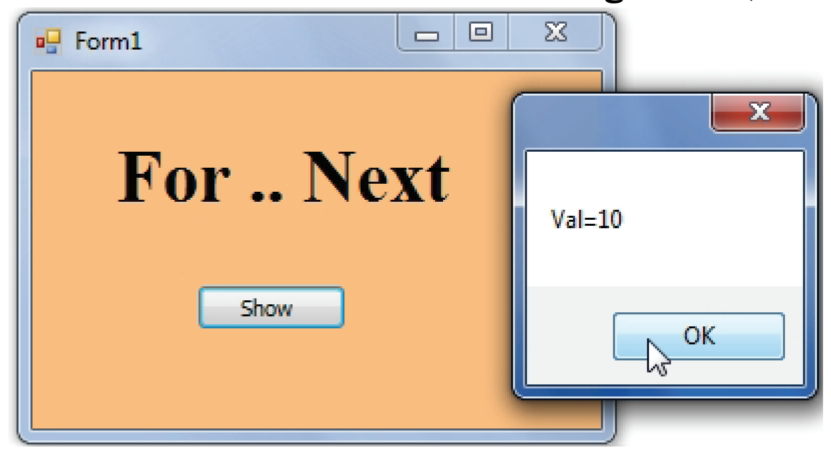
4 – إضغط مرتين على المفتاح Show .. وأكتب الكود التالى :

```
Dim val As Integer
      Dim Start_val As Integer = 1
      Dim End_val As Integer = 10
      For val = Start_val To End_val
         MessageBox.Show("Val=" & Val)
Next
```

5 – شغل البرنامج .. بالضغط على مفتاح **F5** .. ثم إضغط مفتاح Show فتظهر رسالة الحلقة التكرارية For .. Next كما بالشكل التالى :



6 – إضغط **Ok** أكثر من مرة .. فيتم زيادة قيمة المتغير بمقدار واحد حتى تصل لنهاية العد ، وهو الرقم 10 .. وبعدها يتم الخروج من البرنامج .. كما بالشكل التالى :



2 – الحلقة التكرارية Do .. Loop :

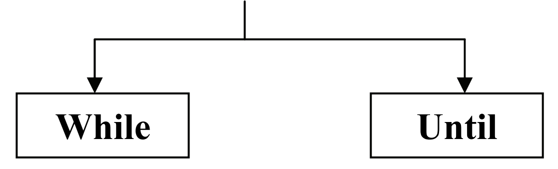
تعمل هذه الحلقة التكرارية على تنفيذ سطر ، أو عدة سطور .. وذلك فى صحة التعبير الذى نقوم بتحديده .. حيث تقوم لغة الـ Visual Basic بحساب هذا التعبير .. فإذا كـان صحيحاً يتم تنفيذ الـسطور الموجودة داخل الحلقة التكرارية .

- وعند الوصول لنهاية الحلقة التكرارية Do .. Loop يتم حساب التعبير مرة أخرى .

- وعند الوصول لنهاية الحلقة التكرارية Do .. Loop يتم حساب التعبير مرة أخرى ، فإذا كان صحيحاً .. يتم تنفيذ مجموعة السطور الموجودة داخل الحلقة التكرارية مرة أخرى .

- وإذا كان خطأ يتم الخروج من الحلقة التكرارية ، والإنتقال لتنفيذ أول سطر موجود بعد الحلقة التكرارية داخل البرنامج .

■ **وتوجد صورتين من الحلقات التكرارية Do .. Loop وهما :**



■ الجملة While :

تستخدم لتكرار سطر أو عدة سطور من الكود طالما أن التعبير Condition صحيح .. وتكون الصيغة العامة للجملة While .. كما بالصيغة التالية :

```
Do While Contition
   Statement Block
Loop
```



صمم برنامج لتكرار الأعداد الزوجية فقط .. أثناء عرض الرسالة عند تشغيل البرنامج .. ولتنفيذ ذلك .. إتبع الخطوات التالية :

1 – إفتح مشروع جديد New Project .

2 – أضف مفتاح Button لنافذة النموذج ،ثم غير الخاصية Text لتصبح Show .

3 – إضغط مرتين على المفتاح Show .. وأكتب الكود التالى :

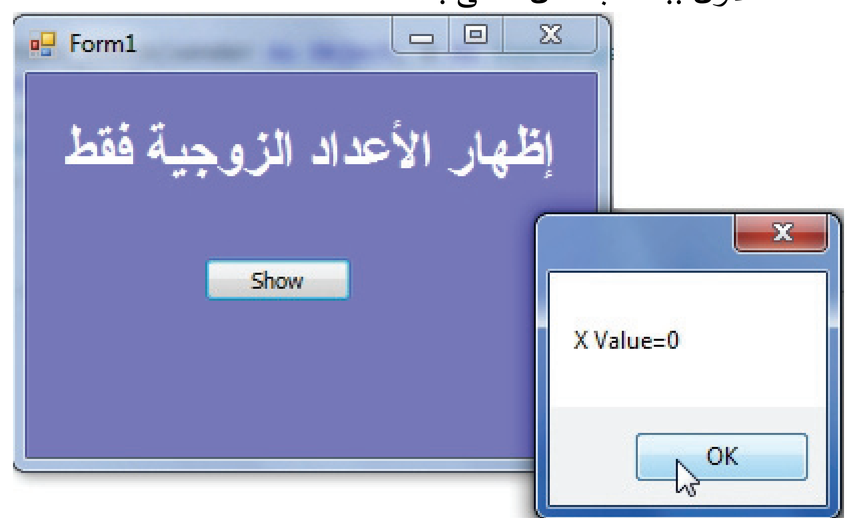
```
Dim X As Integer = 0
 Do While X <= 10
      MessageBox.Show("X Value=" & X)
       X = X + 2
Loop
```

معنى الكود السابق !!

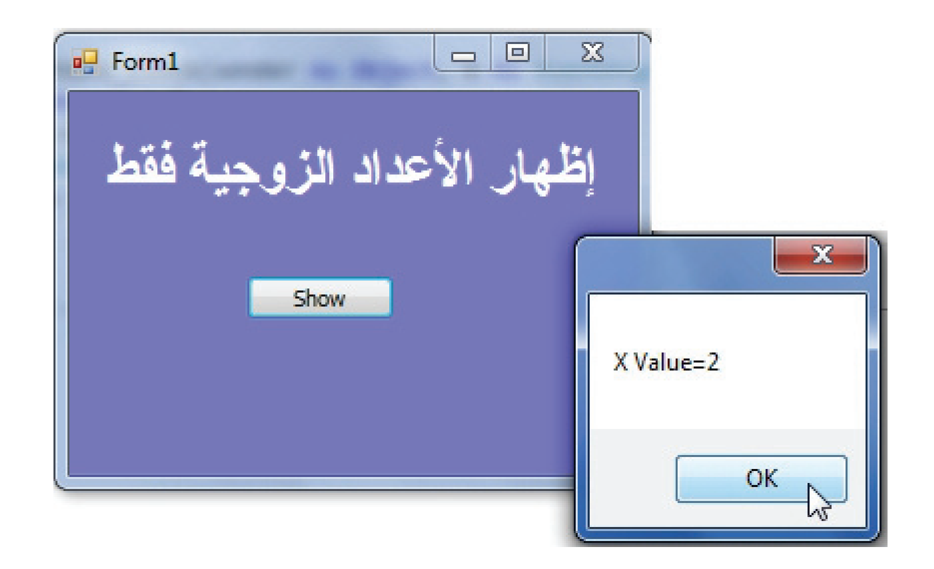
- **أنه تم تعريف متغير من النوع Integr بأسم X ، بحيث يكون أول رقم فى بداية العد أو الإظهار هو الرقم صفر حتى أقل من أو يساوى 10 .**

- **إضافة الرقم 2 للمتغير X .. وذلك للعد برقم زوجى .. أى أن الفرق بين الرقم ، والرقم التالى يكون بمقدار 2 .**

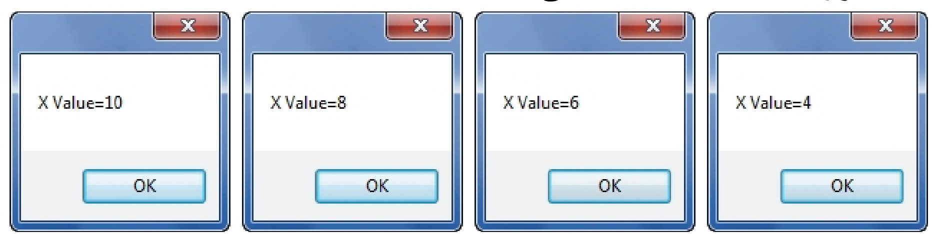
4 – شغل البرنامج .. بالضغط على مفتاح **F5** .. ثم إضغط مفتاح **OK** فيظهر العدد الأول .. كما بالشكل التالى :



5 – إضغط مفتاح **OK** مرة أخرى .. فيظهر العدد الثانى .. كما بالشكل التالى :



6 – كرر الضغط على **OK** .. حتى ينتهى البرنامج من إظهار كل الأعداد الزوجية .. كما بالشكل التالى :



■ الجملة Until :

تستخدم لتكرار تنفيذ السطر ، أو عدة سطور من الكود .. بحيث يصبح التعبير Condition صحيح .. وتكون الصيغة العامة للجملة Until .. كما بالصيغة التالية :

```
Do Until Contition
    Statement Block
Loop
```

ولتنفيذ إستخدام الجملة Until .. من خلال المثال التالى :



نفس المثال السابق .. ولكن إظهار الأعداد الفردية فقط من خـــلال الجملة Do Until .. ولتنفيذ ذلك .. أكتب الكود التالى للمفتاح Show :

```
Dim X As Integer = 1
 Do Until X = 10
        MessageBox.Show("X Value=" & X)
        X = X + 2
Loop
```



تم كتابة الرقم 1 للمتغير X بدلاً من الرقم 0 .. وذلك لإظهار الأرقام الفردية فقط بمقدار 2 .

• شغل البرنامج .. بالضغط على مفتاح **F5** .. ثم إضغط مفتاح **Show** فيظهر أول رقم فردى .. كما بالشكل التالى :



3 – **الحلقة التكرارية While .. End While :**

تعمل هذه الحلقة التكرارية على تنفيذ سطر ، أو عدة سطور من الكود .. طالما أن الشرط الذى نقوم بتحديده صحيح .. وتكون الصيغة العامة .. كما بالصيغة التالية :

```
While Condition
        Statement Block
End While
        Statement 2
```

• فى حالة صحة الشرط Condition .. يتم تنفيذ السطر ، أو مجموعة السطور الموجودة داخل الحلقة التكرارية .

• وعند الوصول لنهاية الحلقة التكرارية .. وهى الجملة End While يتم الإنتقال للجملة While لإختبار الشرط مرة أخرى .. فإذا كان الشرط صحيح يتم تنفيذ مجموعة السطور الموجودة داخل الحلقة التكرارية مرة أخرى .

• أما إذا كان الشرط خطأ .. فيتم الخروج من الحلقة التكرارية ، والإنتقال إلى أول سطر موجود أسفل الجملة While End داخل البرنامج لتنفيذه .

ولتوضيح عمل الحلقة التكرارية While .. End While من خلال المثال التالى :



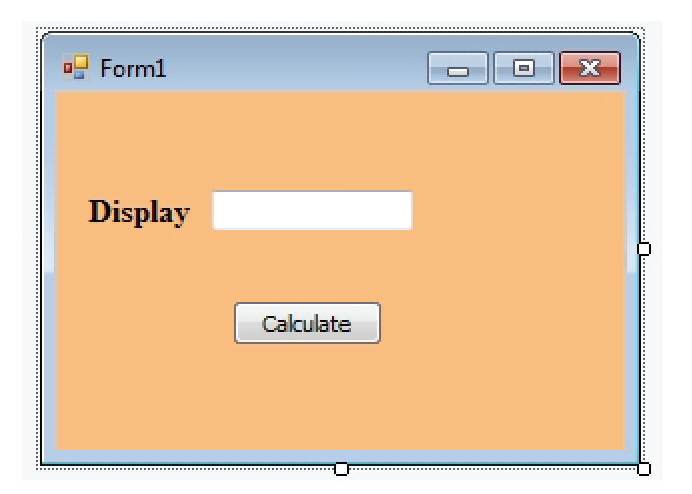
صمم برنامج لحساب مجموعة من الأعداد الموجبة فقط .. ليقوم المستخدم بإدخال القيم من خلال مربع حوارى Input Box يظهر له عند إدخال البيانات بحيث يتم إنهاء البرنامج فى حالة إدخال أى قيمة سالبة فى مربع الإدخال .

ولتنفيذ ذلك .. إتبع الخطوات التالية :

1 – إفتح مشروع جديد New Project .

2 – أضف Text Box ، ومفتاح تحكم Button ، وعنوان Label داخل النموذج Form .

3 – غير الخاصية Text للعنوان Label .. لتصبح Display ، ثم غير الخاصية Text أيضاً للمفتاح Button 1 .. لتصبح Calculate فيظهر النموذج .. كما بالشكل التالى :



4 – إضغط مرتين على المفتاح Calculate .. وأكتب الكود التالى :

```
Dim Number, Total  As Double
Number = 0
 While Number >= 0
        Total = Total + Number
         Number = InputBox ("من فضلك إدخل رقم أخر")
 End While
TextBox1.Text=Total
```



• تم تعريف المتغير Number .. لحفظ القيمة التى يتم إدخالها من خلال مربع الحوار ، والمتغير Total .. لحساب مجموع القيم الموجبة فقط التى يقوم المستخدم بإدخالها ، وكل من المتغيرين من النوع Double .

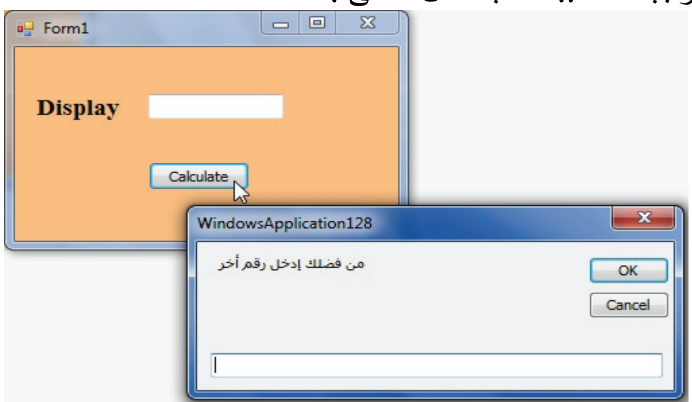
- **الدالة Input Box .. تستخدم مع المتغير Number .. وذلك لتخزين القيمة الموجبة فقط التى يقوم المستخدم بإدخالها .**

- **الجملة Total .. تستخدم لحساب مجموع القيم الموجبة فقط .**

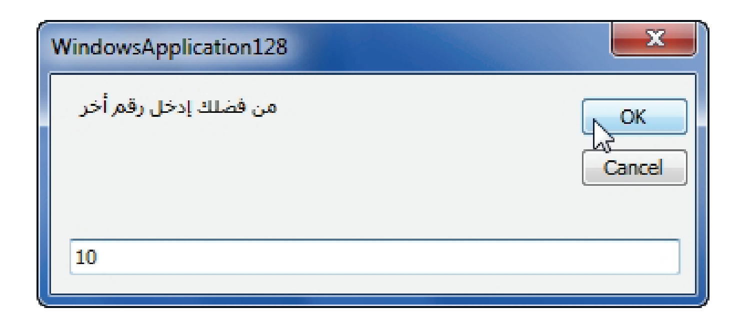
- **الجملة While .. تعنى الرقم أكبر من أو يساوى صفر .. بمعنى إنهاء عملية إدخال الأرقام بمجرد إدخال أى قيمة سالبة .. وبالتالى يتم تنفيذ باقى الجمل الموجودة داخل الحلقة التكرارية .. طالما أن القيم التى يقوم المستخدم بإدخالها موجبة .**

- **وعند إدخال قيمة سالبة .. يتم الخروج من الحلقة التكرارية ، والإنتقال للسطر التالى للحلقة التكرارية داخل البرنامج .. وهو السطر الخاص بإظهار ناتج عملية الجمع .**

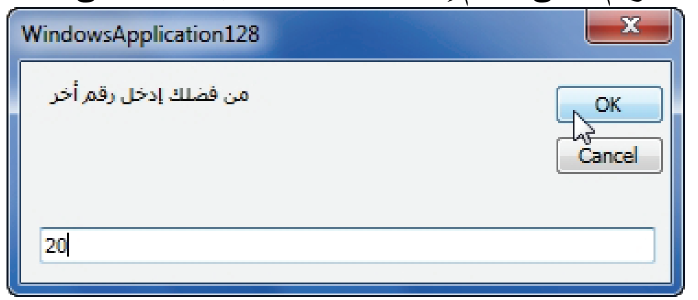
5 – شغل البرنامج .. بالضغط على مفتاح **F5** .. فتظهر نافذة البرنامج ، ثم إضغط مفتاح **Calculate** .. فيظهر مربع الحوار لإدخال الأرقام الموجبة فقط .. كما بالشكل التالى :



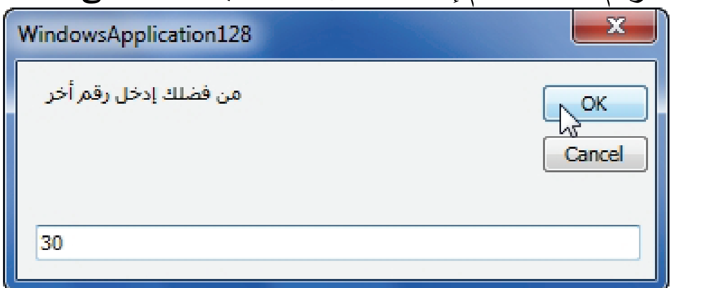
6 – إدخل الرقم الموجب الأول فى مربع الحوار .. ثم إضغط **OK** .. فيظهر مربع الإدخال الثانى .. كما بالشكل التالى :



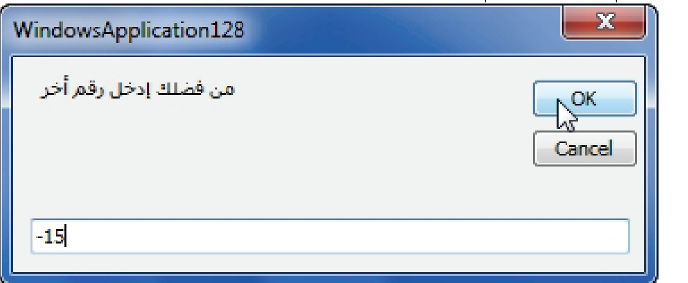
7 – إدخل الرقم الثانى .. ثم إضغط **OK** .. كما بالشكل التالى :



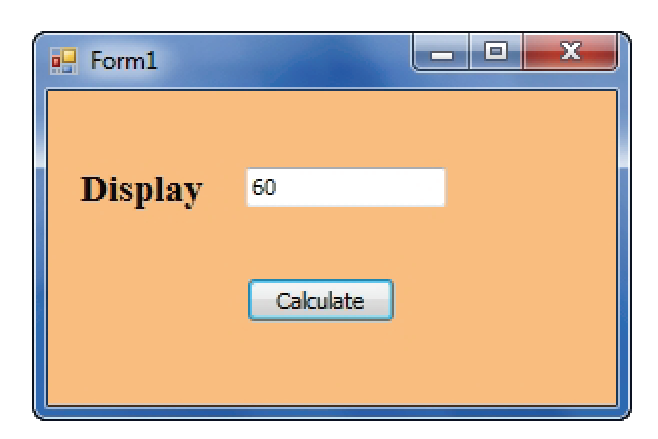
8 – إدخل الرقم الثالث .. ثم إضغط **OK** .. كما بالشكل التالى :



9 – إدخل رقم سالب .. ثم إضغط **OK** .. كما بالشكل التالى :

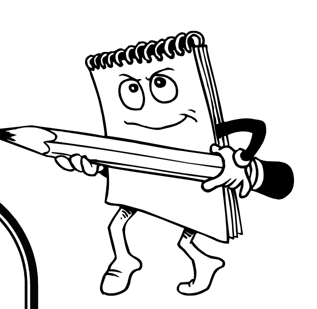


10- يقوم البرنامج بجمع الأرقام الموجبة فقط .. التى تم إدخالها فى مربع الحوار .. وإظهار ناتج الجمع فى المربع Display الموجود بنافذة النموذج Form .. كما بالشكل التالى :



الفصل التاسع

التعامل مع الدوال Functions



نتعلم فى هذا الفصل ..



- ■ الدالة UCase .
- ■ الدالة LCase .
- ■ الدالة StrConv .
- ■ الدالة Len .
- ■ الدالة Replace .
- ■ الدالة StrReverse .
- ■ الدالة Instr .
- ■ الدالة Trim .
- ■ الدالة Beep .
- ■ الدالة Now .
- ■ الدالة Today .

الفصل التاسع

التعامل مع الدوال Functions

نتعلم فى هذا الجزء التعامل مع الدوال التى تتعامل مع المتغير String .. وقـد تـعـلمنا فـى الأجـزاء السابقة بعض الدوال الجاهزة .. مثل الدالة Message Box ، والدالة Input Box .. ولكن فى هذا الفصل نتعامل مع دوال من نوع مختلف .

ولكى يتم توضيح فكرة عمل الدوال التى تتعامل مع المتغير String من خلال مجموعة من الأمثلة .

1 – الدالة UCase :

تستخدم هذه الدالة لتحويل جميع حروف النص المكتوب من حروف صغيرة Small .. إلى حروف كبيرة Capital .. ويمكن توضيح ذلك من خلال المثال التالى :

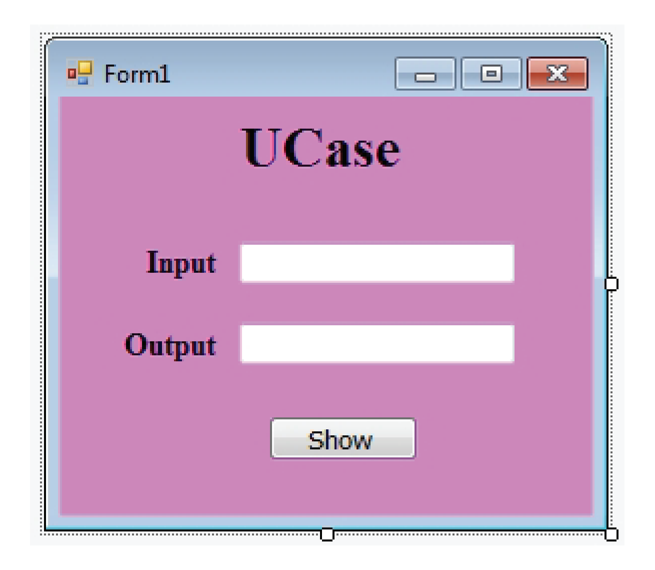


صمم برنامج يقوم بتحويل الحروف الإنجليزية التى يتم إدخالها فى مربع الإدخال من حروف Small إلى حروف Capital .. ولتنفيذ ذلك .. إتبع الخطوات التالية :

1 – إفتح مشروع جديد New Project .

2 – أضف عدد 2 Text Box ، وعدد 3 Lebel للعناوين ، ومفتاح للتحكم Button .. داخل نافذة النموذج Form .

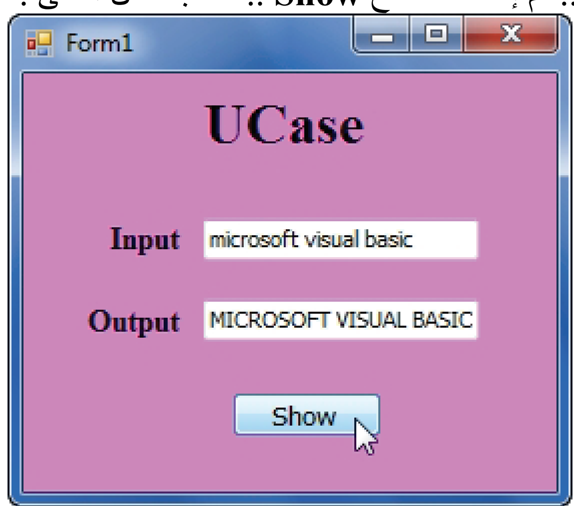
3 – غير الخاصية Text للعنوان الأول لتصبح أسم الدالة (**UCase**) ، والخاصية Text للعنوان الثانى لتصبح (Input) ، والخاصية Text للعنوان الثالث لتصبح (Output) .. ثم غير الخاصية Text لمفتاح التحكم لتصبح (Show) فتظهر نافذة النموذج .. كما بالشكل التالى :



4 – إضغط مرتين على المفتاح Show .. وأكتب الكود التالى :

```
TextBox2.Text=UCase(TextBox1.Text)
```

5 – شغل البرنامج بالضغط على مفتاح **F5** .. ثم أكتب أى نص بحروف Small .. ثم إضغط مفتاح **Show** .. كما بالشكل التالى :



2 – الدالة LCase :

وهذه الدالة عكس الدالة السابقة .. حيث تقوم بتحويل الحروف الكبيرة Capital إلى حروف صغيرة Small .

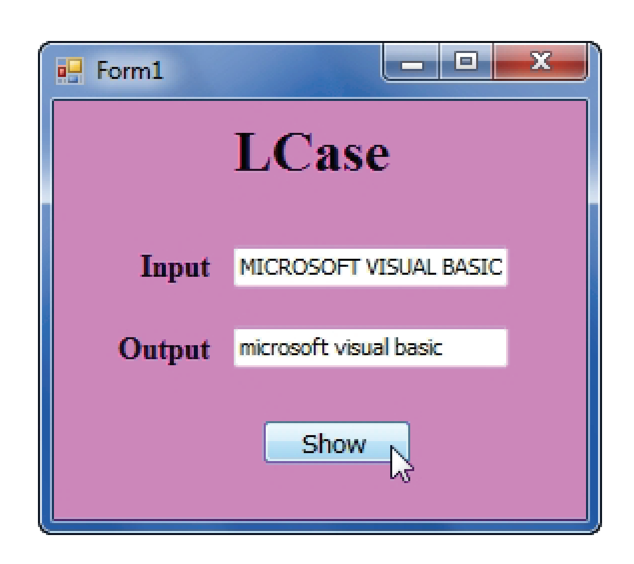


نفس المثال السابق .. ولكن قم بتنفيذ الخطوات التالية :

1 – غيرعنوان النموذج ليصبح (**LCase**) .. ثم إضغط على المفتاح **Show** مرتين وأكتب الكود التالى :

```
TextBox2.Text=LCase(TextBox1.Text)
```

2– شغل البرنامج بالضغط على مفتاح **F5** .. ثم إدخل بيانات بالإنجليزية بحروف كبيرة Capital .. ثم إضغط مفتاح Show .. كما بالشكل التالى :



3 – الدالة StrConv :

تستخدم هذه الدالة لتحويل حروف النص من حالة إلى أخرى .. بمعنى تحويل حروف النص من Capital ، أو Small .



نفس المثال السابق .. ولكن قم بتنفيذ الخطوات التالية :

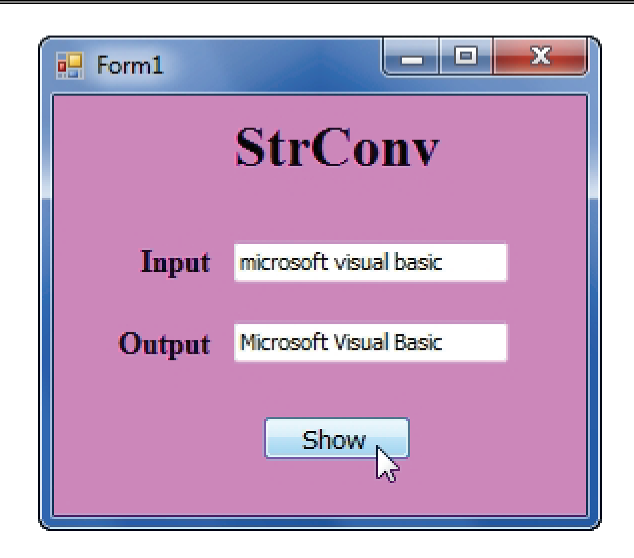
1 – غيرعنوان النموذج ليصبح (**StrConv**) .. ثم إضغط على المفتاح **Show** مرتين وأكتب الكود التالى :

```
TextBox2.Text = StrConv(TextBox1.Text, vbProperCase)
```



الجملة Vb Proper Case فى الكود السابق .. تعنى أن أول حرف من كل كلمة لابد أن يكون Capital ، وباقى حروف النص Small .

2 – شغل البرنامج بالضغط على مفتاح **F5** .. ثم إدخل بيانات بالإنجليزية بحروف صغيرة Small .. ثم إضغط مفتاح **Show** .. كما بالشكل التالى :



4 – الدالة Len :

تستخدم هذه الدالة لإعطاء عدد حروف النص بما فى ذلك المسافات الخالية بين الكلمات .. وهى إختصار للكلمة Lenth .

■ مثال :

نفس المثال السابق .. ولكن قم بتنفيذ الخطوات التالية :

1 – غيرعنوان النموذج ليصبح (Len) .. ثم إضغط على المفتاح **Show** مرتين وأكتب الكود التالى :

```
TextBox2.Text=Len(TextBox1.Text)
```

2 – شغل البرنامج بالضغط على مفتاح **F5** .. ثم إدخل بيانات .. ثم إضغط مفتاح **Show** .. كما بالشكل التالى :



5 – الدالة Replace :

تستخدم هذه الدالة لإستبدال مقطع من النص بمقطع أخر داخل الكود .

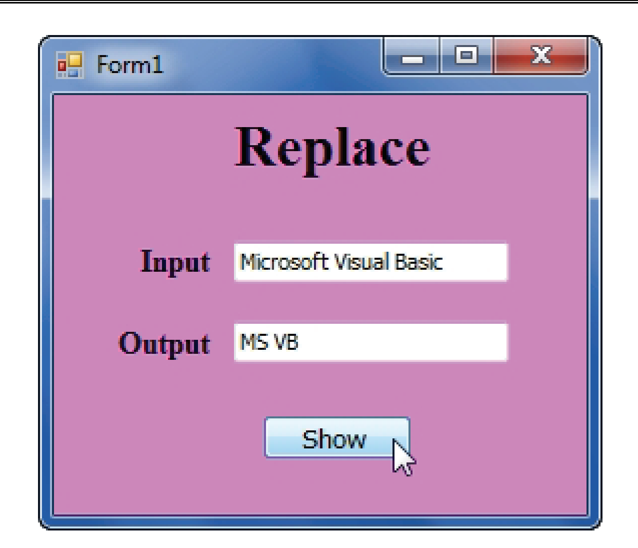


نفس المثال السابق .. ولكن قم بتنفيذ الخطوات التالية :

1 – غيرعنوان النموذج ليصبح (**Replace**) .. ثم إضغط على المفتاح **Show** مرتين وأكتب الكود التالى :

```
TextBox2.Text=Replace(TextBox1.Text, "Microsoft Visual Basic", "MS VB")
```

2 – شغل البرنامج .. بالضغط على مفتاح **F5** .. ثم إدخل الكلمة .. (Microsoft Visual Basic) .. ثم إضغط مفتاح **Show** .. فتظهر النتيجة .. كما بالشكل التالى :





إذا تم وضع علامتى التنصيص " " فارغة داخل الكود .. سوف يتم طباعة نفس النص المكتوب فى المربع Input ، إلى المربع OutPut .. أثناء تشغيل البرنامج .

5 – الدالة StrReverse :

تستخدم هذه الدالة لعكس حروف النص الموجودة داخل مربع الإدخال .. ليظهر كأنه مكتوب فى مرآه .. أى من الخلف للإمام .

▪ مثال :

نفس المثال السابق .. ولكن قم بتنفيذ الخطوات التالية :

1 – غيرعنوان النموذج ليصبح (StrReverse) .. ثم إضغط على المفتاح **Show** مرتين وأكتب الكود التالى :

```
TextBox2.Text=StrReverse(TextBox1.Text)
```

2 – شغل البرنامج بالضغط على مفتاح **F5** .. ثم إدخل أى نص .. ثم إضغط مفتاح **Show** .. كما بالشكل التالى :



6 – الدالة Instr :

تستخدم للبحث عن قيمة معينة داخل النص .

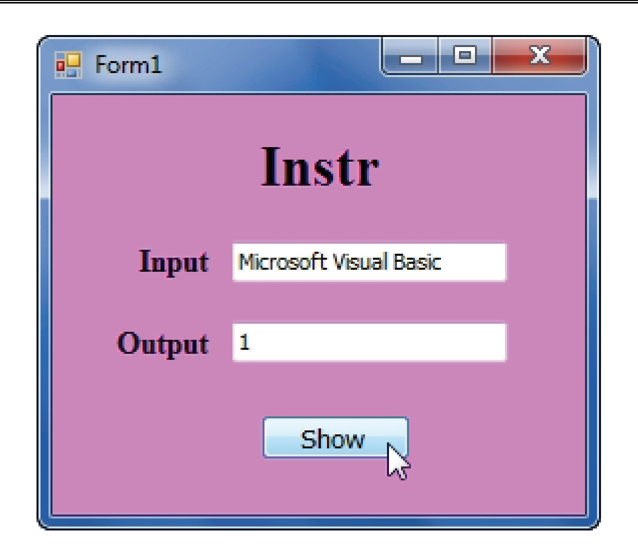


نفس المثال السابق .. ولكن قم بتنفيذ الخطوات التالية :

1 – غيرعنوان النموذج ليصبح **(Instr)** .. ثم إضغط على المفتاح **Show** مرتين وأكتب الكود التالى :

```
TextBox2.Text=InStr(TextBox1.Text, "Microsoft Visual Basic")
```

2 – شغل البرنامج بالضغط على مفتاح **F5** .. ثم إدخل الكلمة .. (Microsoft Visual Basic) .. ثم إضغط مفتاح **Show** .. فتظهر النتيجة .. كما بالشكل التالى :





الرقم 1 .. يعنى أن البرنامج يبحث داخل النص المكتوب عن ترتيب هذا المقطع بداية من الرقم 1 ، وإذا لم تجد الدالة النص المطلوب البحث عنه ستكون القيمة صفر .

7 – الدالة Trim :

تستخدم لحذف المسافات الموجودة يمين ، ويسار النص المكتوب .

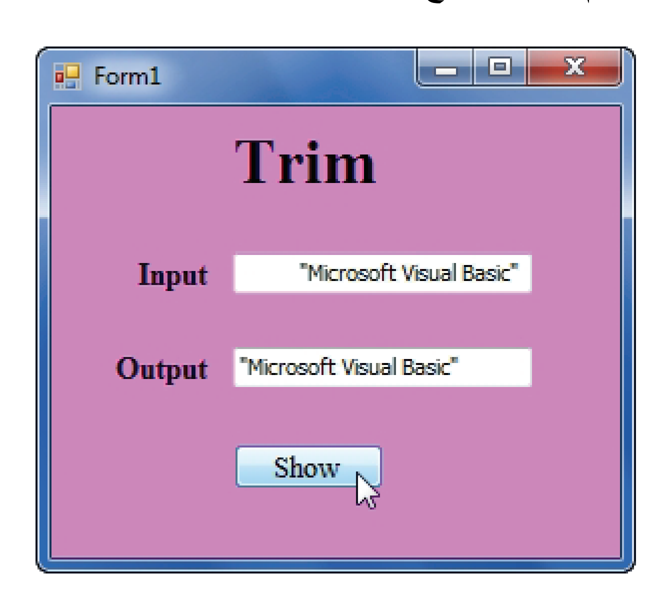


نفس المثال السابق .. ولكن قم بتنفيذ الخطوات التالية :

1 – غيرعنوان النموذج ليصبح **(Trim)** .. ثم إضغط على المفتاح **Show** مرتين وأكتب الكود التالى :

```
TextBox2.Text=Trim(TextBox1.Text)
```

2 – شغل البرنامج بالضغط على مفتاح **F5** .. ثم إدخل الكلمة .. (Microsoft Visual Basic) .. مع ترك مسافات قبل بداية كتابة الكلمة ثم إضغط مفتاح **Show** .. فتظهر النتيجة .. كما بالشكل التالى :





لحذف المسافات يمين النص المكتوب فقط نستخدم الدالة RTrim ، ولحذف المسافات يسار النص المكتوب نستخدم الدالة LTrim .. بنفس صيغة الدالة السابقة Trim .

8 – الدالة Beep :

تستخدم لإصدار صوت تنبيهى أثناء الضغط على أى مفتاح من المفاتيح الموجودة داخل النموذج Form .. مثل مفتاح الحفظ ، أو الإغلاق .. يمكن تنفيذ الدالة Beep .. وذلك بكتابة الكود التالى للمفتاح المطلوب إصدار الصوت منه :

```
Interaction.Beep()
```

9 – الدالة Now :

تستخدم لإظهار والوقت ، والتاريخ الحالى .. وهى مرتبطة بالوقت والتاريخ الخاص بجهاز الكمبيوتر .. وتستخدم هذه الداله فى نماذج الفواتير ، وعمليات البيع ، والبنوك .



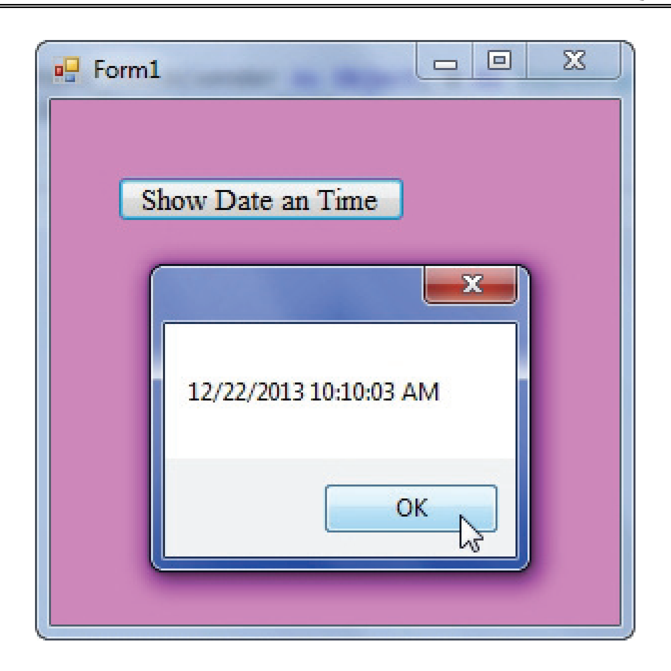
صمم برنامج يحتوى على مفتاح بمجرد الضغط عليه تظهر رسالة بها الوقت والتاريخ .. ولتنفيذ ذلك .. إتبع الخطوات التالية :

1 – أضف مفتاح تحكم Button .. للنموذج Form .. ثم غير الخاصية Text لتصبح Show Date and Time .

2 – إضغط مرتين على المفتاح Show Date and Time .. وأكتب الكود التالى :

```
MessageBox.Show(DateAndTime.Now)
```

3– شغل البرنامج بالضغط على مفتاح **F5** .. ثم إضغط مفتاح .. Show Date and Time .. فتظهر رسالة بها الوقت والتاريخ الحالى .. كما بالشكل التالى :



10 – الدالة Today :

تستخدم لإظهار تاريخ اليوم فقط .



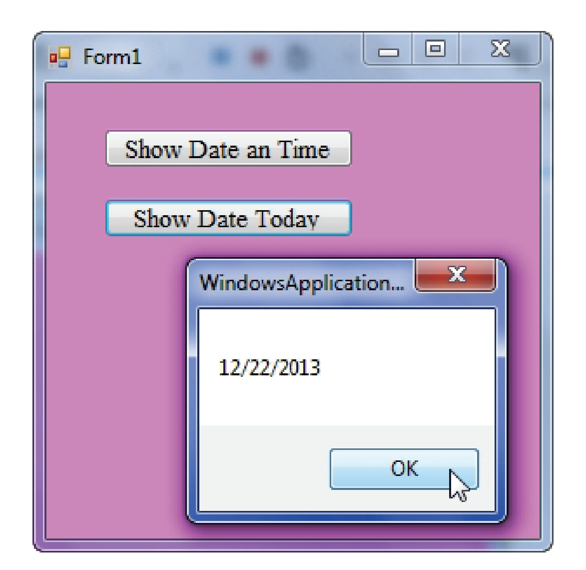
نفس المثال السابق .. ولكن قم بتنفيذ الخطوات التالية :

1 – أضف مفتاح تحكم Button .. للنموذج Form .. ثم غير الخاصية Text لتصبح Show Date Today .

2 – إضغط مرتين على المفتاح Show Date Today.. وأكتب الكود التالى :

```
MessageBox.Show(DateAndTime.Today)
```

3– شغل البرنامج بالضغط على مفتاح **F5** .. ثم إضغط مفتاح .. Show Date Today .. فتظهر رسالة بها تاريخ اليوم فقط .. كما بالشكل التالى :



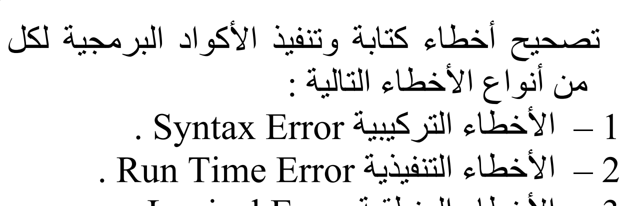


يمكن إستخدام الدالة **MsgBox** .. بدلاً من الدالة **MessageBox** .. وتكون كما بالصيغ التالية :

```
MsgBox(DateAndTime.Now)
```

```
MsgBox(DateAndTime. Today)
```

الفصل العاشر
إكتشاف وتصحيح الأخطاء



نتعلم فى هذا الفصل ..

تصحيح أخطاء كتابة وتنفيذ الأكواد البرمجية لكل من أنواع الأخطاء التالية :

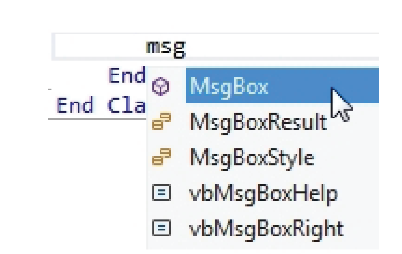
1 – الأخطاء التركيبية Syntax Error .
2 – الأخطاء التنفيذية Run Time Error .
3 – الأخطاء المنطقية Logical Error .

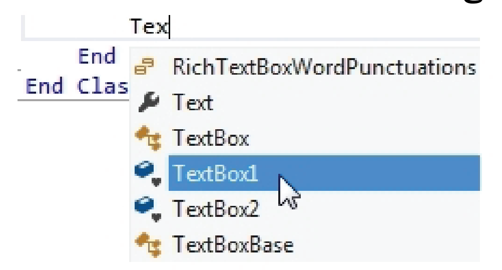
الفصل العاشر
إكتشاف وتصحيح الأخطاء

الخطأ فى لغة الـ Visual Basic يسمى Bug .. وتصحيح الأخطاء يسمى Debug .

• **فالسؤال .. كيف يمكننا كتابة برنامج خالى من الأخطاء ؟**
لكتالة برنامج خالى من الأخطاء داخل لغة الـ Visual Basic عليك أن تنفذ الخطوات التالية :
1 – جرب تنفيذ الأوامر Testing للتأكد من أنها تعمل بالشكل الصحيح .

2 – تتبع كتابة الكود فى صفحة محرر الأكواد Code Editor أثناء الكتابة الديناميكية للكود .. كما بالشكل التالى :





3 – إفحص البرنامج Debugging عن طريق تنفيذ سلسلة من التجارب للتأكد من أن البرنامج ككل يعمل بالشكل المطلوب .. وذلك عن طريق :

• تجربة جزء من البرنامج .. خصوصاً إذا كان البرنامج يتكون من مجموعة كبيرة من الأجزاء .

• تجربة كل أمر تتأكد من عمله .. مع ملاحظة أن هذ لا يعنى أن البرنامج سيعمل بالطريقة المطلوبة .

4 – إجراء سلسلة من عمليات الفحص للتأكد من أن البرنامج سيعمل كما هو مخطط له .. من خلال وصف دورة تصميم البرامج .. وهى :
• دراسة وفهم المشكلة .

- تصميم الحل .
- كتابة البرنامج .
- فحص وتجربة البرنامج .

5 – تجربة البرنامج بإستخدام بعض البيانات الإفتراضية .. ويحب تجربة هذه الخطوة عدة مرات للتأكد من عمل البرنامج .

6 – إستخدم بعض البيانات غير المنطقية لمعرفة رد فعل البرنامج تجاه هذه العمليات مثل العمليات الحسابية باقسمة على صفر .

7 – بعد الإنتهاء من كتابة كل جزء من البرنامج .. حاول تجربته لتتبع الأخطاء بكل جزء .. وهذا أفضل من تجربة البرنامج بالكامل .

8 – قم بتجزئه البرنامج إلى أقسام .. لتتمكن من إكتشاف وتصحيح الأخطاء بطريقة أفضل .



أى مبرمج سواء كان خبير ، أو مبتدئ .. يقع ويرتكب أخطاء أثناء كتابة البرنامج .. فلا تقلق أثناء إرتكابك الأخطاء .. ولكن تعلم من هذه الأخطاء لتفاديها ومعالجتها فيما بعد .

■ أنواع الأخطاء فى لغة الـ Visual Basic :

وتنقسم الأخطاء داخل الـ Visual Basic إلى 3 أنواع .. وهى :

1 – الأخطاء التركيبية Syntax Error .
2 – الأخطاء التنفيذية Run Time Error .
3 – الأخطاء المنطقية Logical Error .

1 – الأخطاء التركيبية Syntax Error :

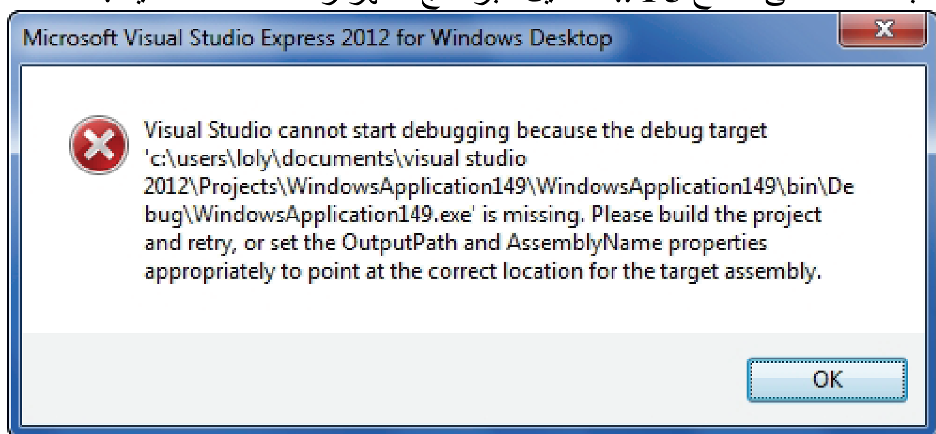
وهى أخطاء مخالفة قواعد كتابة أوامر اللغة .. كما بالكود التالى :

```
Public Class Form1

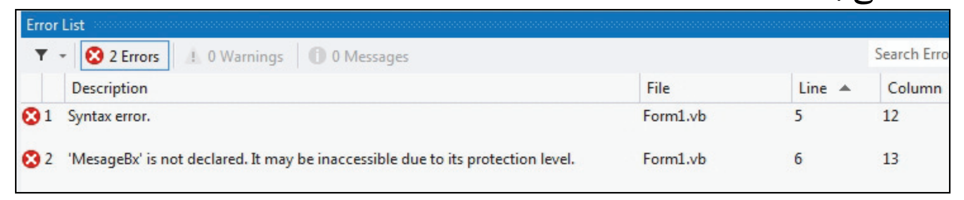
    Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        Dim X As Integer = 0
        Do Whil X <= 10
            MesageBx.Show("X Value=" & X)
            X = X + 2
        Loop
    End Sub
End Class
```

• نجد فى الكود السابق .. أن هناك أخطاء حرفية أثناء كتابة جمل الكود .. مثل الجملة **While** .. مكتوبة **Wil** ، والجملة **MessageBox** .. مكتوبة **MesageBx** .

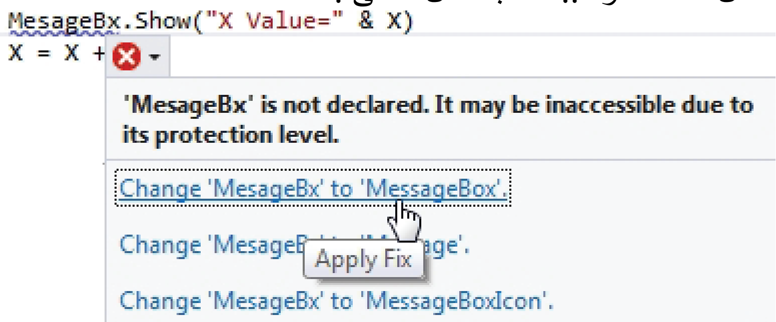
• بالضغط على مفتاح **F5** .. لتشغيل البرنامج تظهر رسالة الخطأ التالية :



• ونتتبع القائمة Error List التى تظهر أسفل نافذة كتابة الكود .. كما بالشكل التالى :



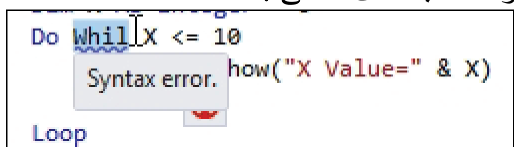
• إضغط على الخطأ الأول .. الموجود فى قائمة الأخطاء .. فيتم تحديد الخطأ داخل نافذة الكود .. كما بالشكل التالى :





بعد الضغط على أسم الخطأ الأول فى قائمة الأخطاء .. حدد لنا البرنامج أسم ومكان الخطأ .. وكذلك تصحيح الخطأ .. وهى كلمة <u>MessageBox</u> .. بدلاً من الكلمة MesgeBx .

• إضغط على الخطأ الثانى .. فى القائمة Error List .. فيتم تحديد الخطأ داخل نافذة الكود كما بالشكل التالى :





ظهر نوع الخطأ فى نافذة كتابة الكود .. وهو Syntax Error أى خطأ مخالفة كتابة قواعد اللغة .. قم بتصحيح الكلمة <u>While</u> بدلاً من <u>Whil</u> .

2 – **الأخطاء التنفيذية Run Time Error :**

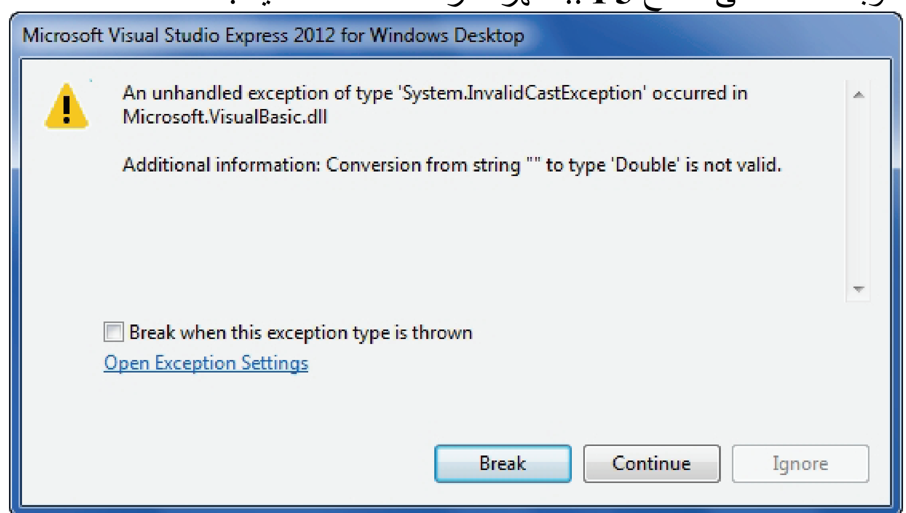
وهو الخطأ الذى ليس له علاقة بصحة كتابة الكود .. ولكنه يظهر عندما يطلب من البرنامج أداء وظيفة معينة يستحيل على البرنامج إجرائها .. مثل القسمة على الرقم صفر ، أو إستدعاء صورة من ملف .. وهذا الملف غير موجود على جهاز الكمبيوتر .

وبالتالى يظهر هذا الخطأ بعد تنفيذ البرنامج .. وعندما يصل الكود إلى هذه النقطة تظهر رسالة الخطأ .

- فمثلاً .. ظهر الكود التالى ، وبه معادلة بالقسمة على صفر ..

```
TextBox3.Text = Val(TextBox1.Text) * Val(TextBox2.Text * 2.5 / 0)
```

- وبالضغط على مفتاح **F5** .. ظهرت رسالة الخطأ التالية :



- بالضغط على مفتاح **Break** .. يتم العودة إلى بداية سطر الكود فى نافذة كتابة الكود .

- وبالضغط على مفتاح **Continue** .. يتم تشغيل البرنامج بالشكل الخاطئ وتجاهل الأخطاء .

3 – الأخطاء المنطقية Logical Error :

وهى من أصعب الأنواع من حيث إكتشافه .. وهو ليس خطأ فى كتابة الكود ، وليس خطأ فى عمليات البرنامج .. ولكن هو خطأ ناتج عن سوء التقدير ، أو السهو والنسيان من قبل المبرمج فى كتابة الكود .

• **فمثلاً :**

عند إجراء عملية حسابية داخل الكود البرمجى .. لإجراء طرح الحوافز من المرتب الأساسى .. وبالتالى يكون الناتج خطأ .. لأنه من المفروض جمع الحوافز على المرتب الأساسى .

وبالتالى يكتشف هذا الخطأ عندما نحصل على ناتج غير متوقع من البرنامج ويلزم هنا إعادة إختبار البرنامج مرة أخرى ، والتدقيق فى نتائجة للوصول إلى مكان الخطأ وتصحيحة .

• ويمكن التغلب على هذا النوع من الأخطاء بتقسيم البرنامج إلى عدة أجزاء .. بحيث يكون كل جزء مسئول عن مهمة معينة .. وبالتالى يساعدنا على معرفة الخلل فى أى جزء من البرنامج .

• ولمتابع الخطأ بإستخدام Break Mode .. من القائمة Dbug إختار الأمر Toggle Break Point ، أو بالضغط على مفتاح **F9** .. فيتم تظليل الكود بلون مختلف ووضع دائرة لونها أحمر عند بداية سطر الكود .. كما بالشكل التالى :

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
    TextBox3.Text = Val(TextBox1.Text) * Val(TextBox2.Text  2.5 / 0)
End Sub
End Class
```



هناك أخطاء تظهر عند كتابة الكود البرمجى .. وهناك أخطاء أخرى لن تظهر إلا عند تشغيل البرنامج .. وهناك أخطاء أخرى تكتشف بالصدفة بعد تشغيل وإستخدام البرنامج فترة .

الفصل الحادى عشر
قواعد البيانات
Data Base

نتعلم فى هذا الفصل ..



- تعريف قواعد البيانات Data Base .
- مكونات قاعدة البيانات .
- برامج قواعد البيانات .
- تصميم قاعدة بيانات للموظفين بإستخدام برنامج Microsoft Access .
- تصميم وجهات التطبيق ، وربطها بقاعدة البيانات .
- الإتصال بحقول قاعدة البيانات .
- إخراج البرنامج للعمل من خلال أيقونة Setup .

الفصل الحادى عشر

قواعد البيانات

Data Base

تعتبر قواعد البيانات من العمليات الهامة فى أى نظام معلومات لأى منظمة أعمال .. بل لها أهمية كبيرة أيضاً فى حياتنا اليومية ..

• **فالسؤال .. ماهى قواعد البيانات ؟**

■ قواعد البيانات Data Base :

هى تجميع منظم لجميع الملفات المتشابهة فى مكان واحد مركزى .. مثل قواعد بيانات (الموظفين / الطلاب / العملاء / الموردين / المخازن / دليل التليفون / بطاقات الرقم القومى / جوزات السفر) وغيرها من الملفات والبيانات المترابطة .

■ مكونات قاعدة البيانات Data Base :

تتكون قاعدة البيانات من مجموعة من الجداول ، أو الملفات .. والتى لها مجموعة من الخصائص .. ويتكون كل جدول ، أو ملف داخل قاعدة البيانات على الأتى :

1 – الحقل Field :

وهو يمثل عمود الجدول .. ويحتوى على عنصر أساسى لملف قاعدة البيانات مثل .. (الأسم ، والعنوان ، ورقم التليفون) .. كما بالشكل التالى :

رقم الموظف	أسم الموظف
1301	وسيم هدهد
1317	نادر نور
1300	السيد أبو حفيظة
1315	احمد شوقى
1305	وليد وحيد
1311	على عمر
1307	تامر سمير

2 – السجل Record :

وهو يمثل صف الجدول .. وهو يعبر عن بيانات كل شخص بالتفصيل .. مثل (أسم الشخص نفسه / عنوانه الفعلى / رقم التليفون) .. كما بالشكل التالى :

رقم الموظف	أسم الموظف	رقم التليفون	المرتب	رقم القسم
1301	وسيم هدهد	01112223334	$1,250.00	1

3 – الخاصية Attribute :

وهى تمثل خلايا الجدول .. وهى الصفات ، أو المعلومات الأساسية للكيان .. مثل رقم الموظف 1310 .. وهو محور البحث داخل جدول قاعدة البيانات .

- **إذن** .. تتكون قاعدة البيانات من مجموعة من الملفات ، والجداول تربط بينهم علاقات Relationships .. ويتم العمل داخل قاعدة البيانات من خلال واجهة المستخدم النهائية End User Interface .. وهو محور عمل هذا الفصل داخل لغة الـ Visual Basic .

■ برامج قواعد البيانات Data Base :

هناك العديد من برامج قواعد البيانات .. وكل برنامج حسب إمكانيات وإحتياجات كل منظمة ، أو شركة .. مثل قواعد بيانات :

- Microsoft Access .
- Microsoft SQL Server .
- ORACLE .

وسنتناول فى هذا الفصل .. تصميم قاعدة بيانات بسيطة للموظفين Employee .. بإستخدام برنامج Microsoft Access .. وتنفيذ واجهة التطبيق (الإستخدام) داخل لغة الـ Visual Basic .

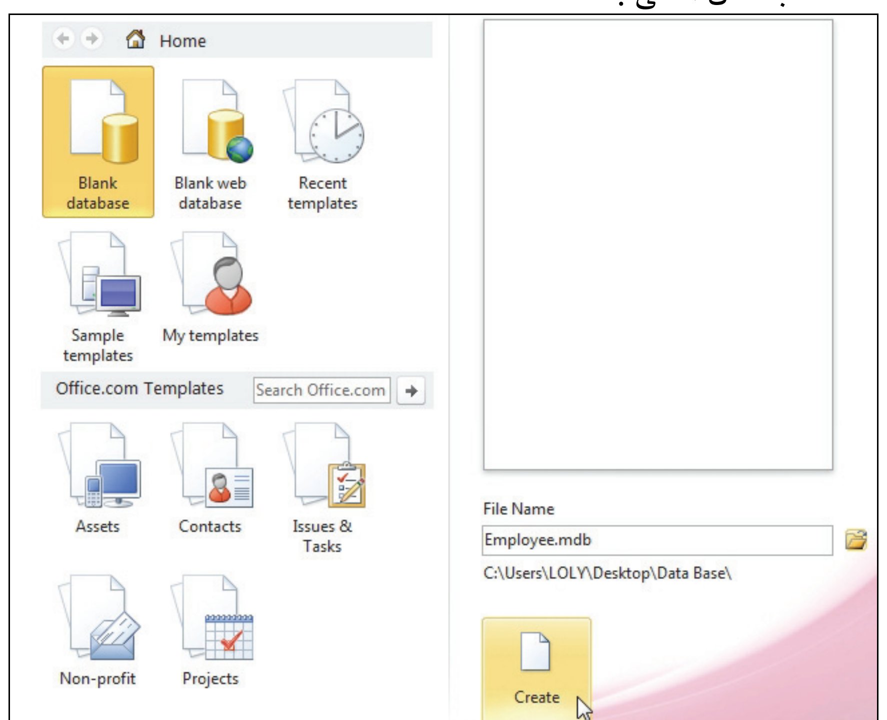
- **ولتنفيذ تصميم وإدارة قواعد البيانات .. من خلال 3 مراحل كالأتى :**
  - المرحلة الأولى : إنشاء قاعدة البيانات .
  - المرحلة الثانية : تصميم واجهة التطبيق وربطها بقاعدة البيانات .
  - المرحلة الثالثة : إخراج البرنامج للعمل من خلال أيقونة Setup .

■ المرحلة الأولى : إنشاء قاعدة البيانات :

■ أولاً : إنشاء ملف قاعدة البيانات Create Database File :

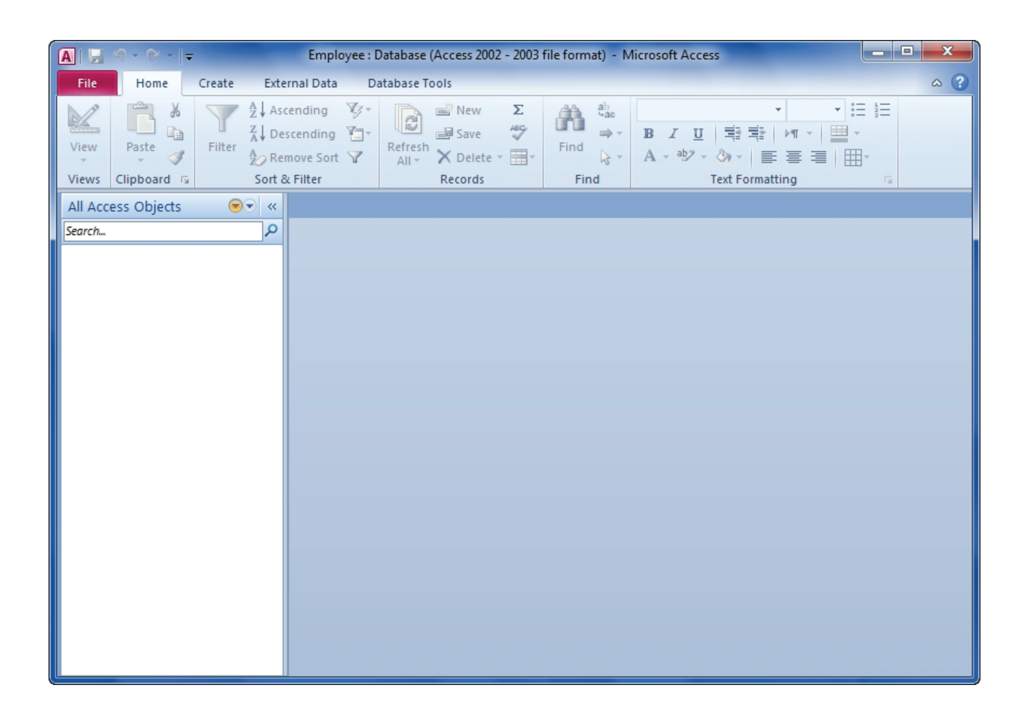
1 – قم بـإنشاء مـجلد Folder فى أى مكان تريده على جهاز الكمبيوتر بأسم Bata Base .. وسوف يوضع بداخله ملف قاعدة البيانات ، والبرنامج الذى يتم تصميمة بلغة Visual Basic .

2 – إفتح برنامج Microsoft Access .. فتظهر النافذة الإفتتاحية للبرنامج كما بالشكل التالى :



3 – حدد أسم قاعدة البيانات فى المستطيل File Name .. بأسم الموظفين Employee .. ثم حدد مكان حفظ قاعدة البيانات .. وذلك بالضغط على علامة المجلد الموجودة بجانب المستطيل File Name .. ثم إختار المجلد Data Base .. الذى تم إنشاؤه فى الخطوة السابقة .

4 – إضغط مفتاح Create الموجود فى نافذة واجهة برنامج Access فتظهر أول نافذة للبرنامج .. كما بالشكل التالى :



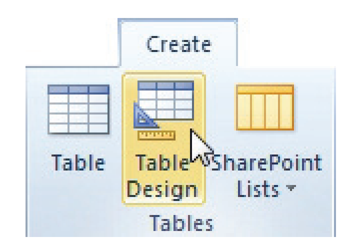
وبعد نشاء الملف الرئيسى لقاعدة البيانات .. نقوم الآن بإنشاء أول جدول فى قاعدة البيانات .

- حيث أن .. **الجداول Tables** تمثل الهيكل الرئيسى لأى قاعدة بيانات ، فهى تمثل المخزن الذى تحفظ بداخله البيانات .

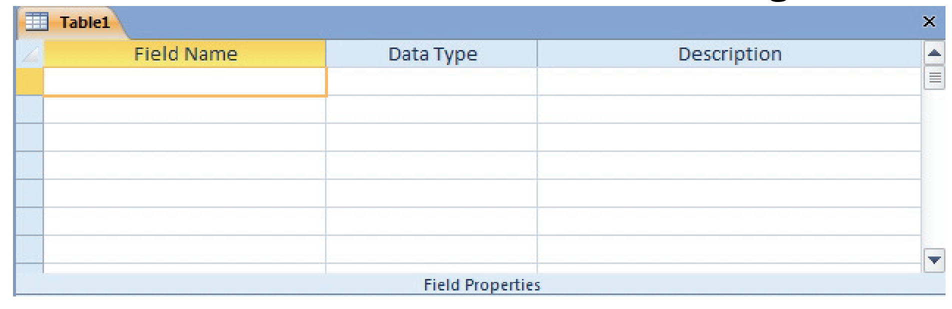
■ ثانياً : إنشاء الجداول Create Tebles :

- **إنشاء الجدول الأول فى قاعدة البيانات وهو جدول الموظفين Employee :**

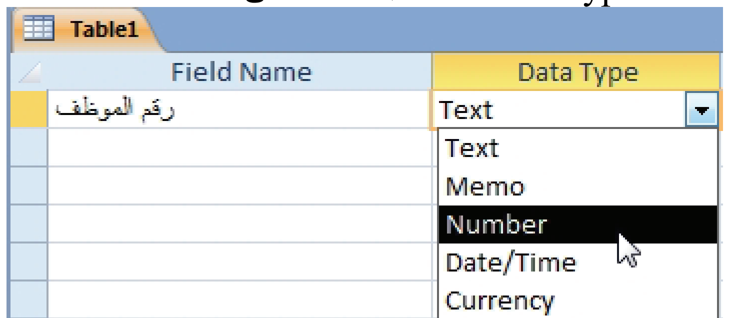
1 – من القائمة Create .. إضغط على الأداة Table Design ..



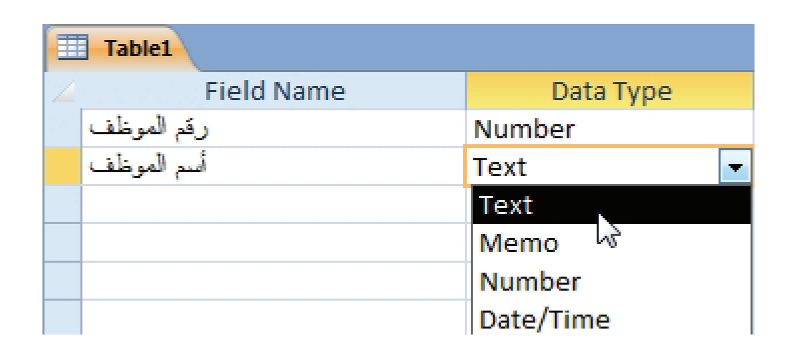
فيظهر هيكل إنشاء الجداول .. وهو الجدول الخاص ببيانات الموظفين كما بالشكل التالى :



2 – قم بإدخال أسم أول حقل فى الجدول فى الخلية Field Name .. وهو رقم الموظف .. ثم إختار نوع الحقل رقمى Number من القائمة المنسدلة Data Type .. كما بالشكل التالى :



3 – إنتقل للصف الثانى فى هيكل الجدول .. وذلك لإدخال الحقل الثانى وهو أسم الموظف .. ونوع الحقل نصى Text .. كما بالشكل التالى :





يتم إختار نوع الحقل من القائمة Data Type .. حسب الغرض من الحقل سواء كان نصى ، أو رقمى ، أو تاريخ .. وإذا لم نحدد نوع الحقل يفترض البرنامج تلقائياً أن نوع الحقل نص Text .

4 – إنتقل للصف الثالث فى هيكل الجدول .. وذلك لإدخال الحقل الثالث وهو العنوان .. ونوع الحقل Text .

5 – إنتقل للصف الرابع فى هيكل الجدول .. وإدخل حقل رقم التليفون .. ونوع الحقل يكون Text .



حقل رقم التليفون يكون من النوع النصى Text حتى يتم قراءة الرقم صفر على الشمال ، وهو رقم بداية أرقام التليفونات (010 ، 011 ، 012) .

6 – إنتقل للصف الخامس فى هيكل الجدول .. وإدخل حقل المرتب .. ونوع الحقل يكون Currency أى عمله .

7 – إنتقل للصف السادس .. وإدخل رقم القسم .. ونـوع الحقل رقمى Number .

• فيظهر الشكل النهائى لهيكل الجدول الأول فى قاعدة البيانات .. كـما بالشكل التالى :

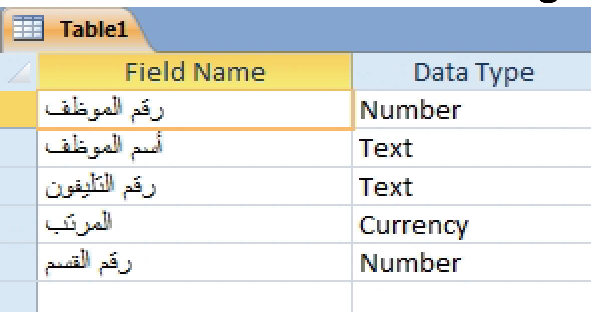
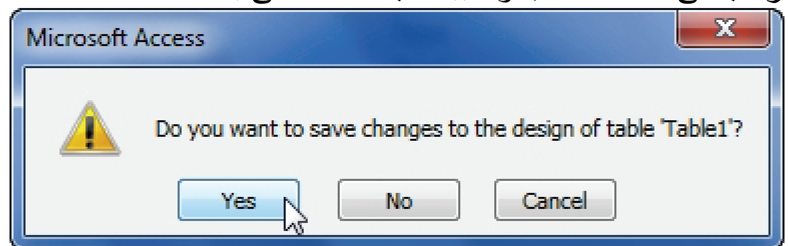


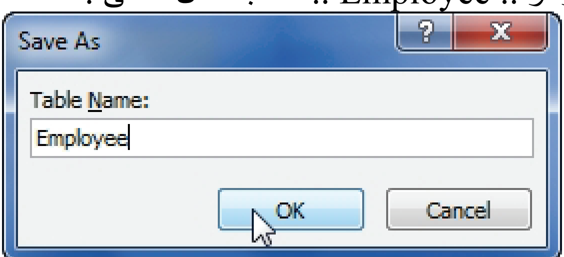
Table1

Field Name	Data Type
رقم الموظف	Number
أسم الموظف	Text
رقم التليفون	Text
المرتب	Currency
رقم القسم	Number

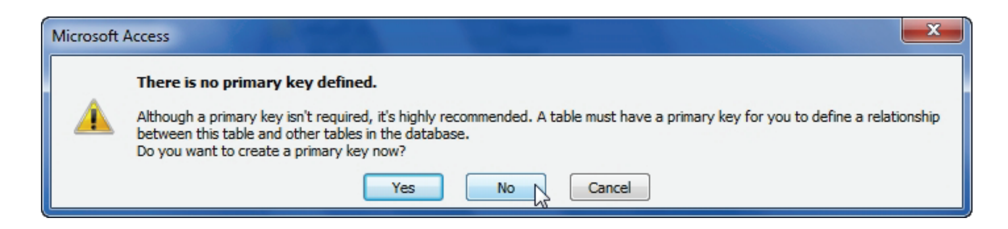
8 – إغلق هيكل الجدول السابق .. وذلك بالضغط على علامة الإغلاق × الموجود أعلى نافذة هيكل الجدول .. فتظهررسالة لتسألك .. هل ترغب فى حفظ هذا الجدول ..كما بالشكل التالى :



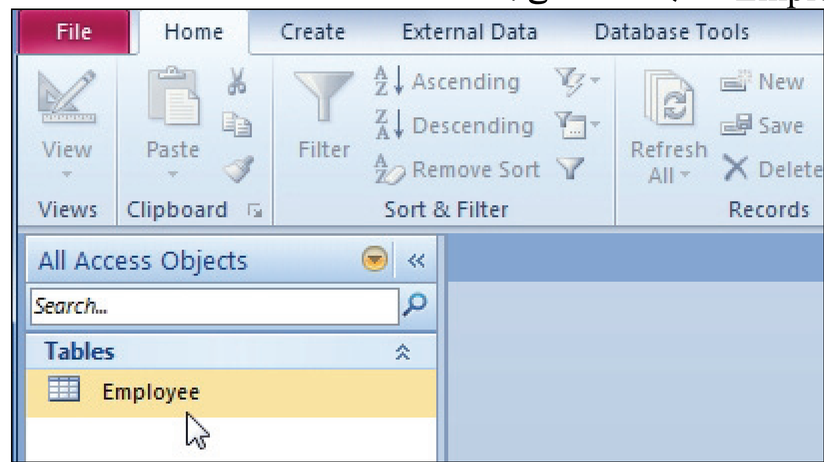
9 – إضغط Yes .. فيظهر مستطيل لتحديد أسم الجدول .. أكتب أسم الجدول الأول وهو .. Employee .. كما بالشكل التالى :



10 – إضغط **OK** .. فتظهر الرسالة التالية :



حيث يسألنا البرنامج .. هل ترغب فى عمل مفتاح أساسى Primary Key لهذا الجدول .. إضغط **No** .. فيظهر أول جدول فى قاعدة البيانات بأسم Employee كما بالشكل التالى :



• **إنشاء الجدول الثانى فى قاعدة البيانات وهو جدول الأقسام Department :**

1 – نفس خطوات إنشاء الجدول السابق .. ثم قم بإدخال حقول الجدول .. كما بالشكل التالى :

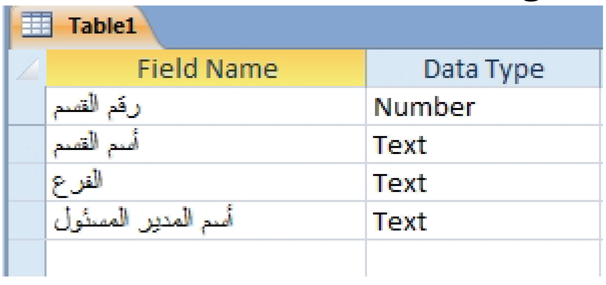
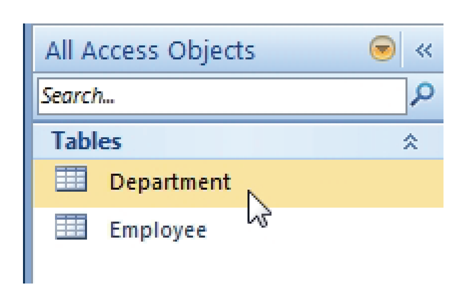


Table1

Field Name	Data Type
رقم القسم	Number
أسم القسم	Text
الفرع	Text
أسم المدير المسئول	Text

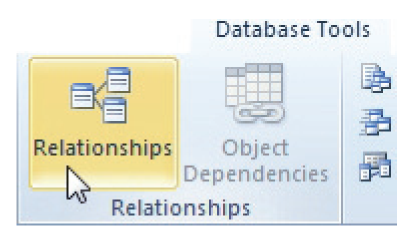
2 – إحفظ الجدول الثانى .. بأسم Department .. فيظهر الجدول الثانى فى قاعدة البيانات .. كما بالشكل التالى :



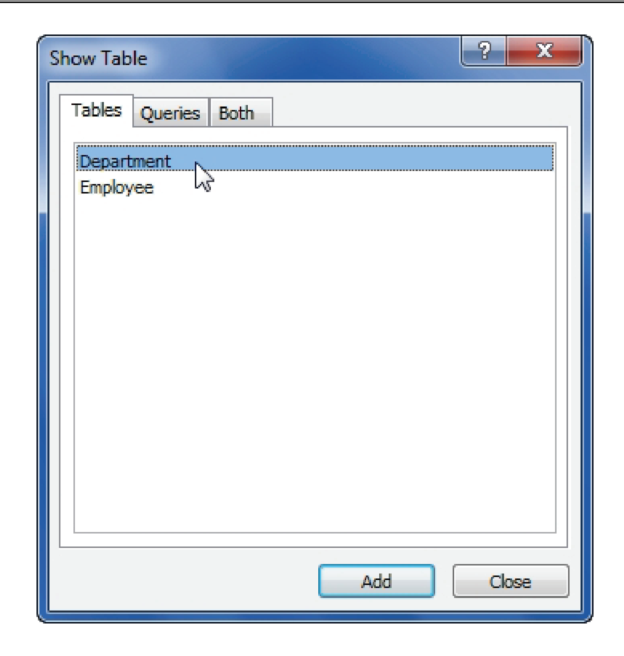
■ ثالثًا : إنشاء العلاقات بين الجداول Relationships :

لكى يحدث تكامل بين الجدولين لابد من وجود علاقة مشتركة بين الجدولين وذلك من خلال حقل مشترك بين الجدولين .. وهو حقل رقم القسم .. ولتنفيذ إنشاء العلاقات بين الجداول .. إتبع الخطوات التالية :

1 – من القائمة Database Tools .. إضغط على الأداة Relationships ..



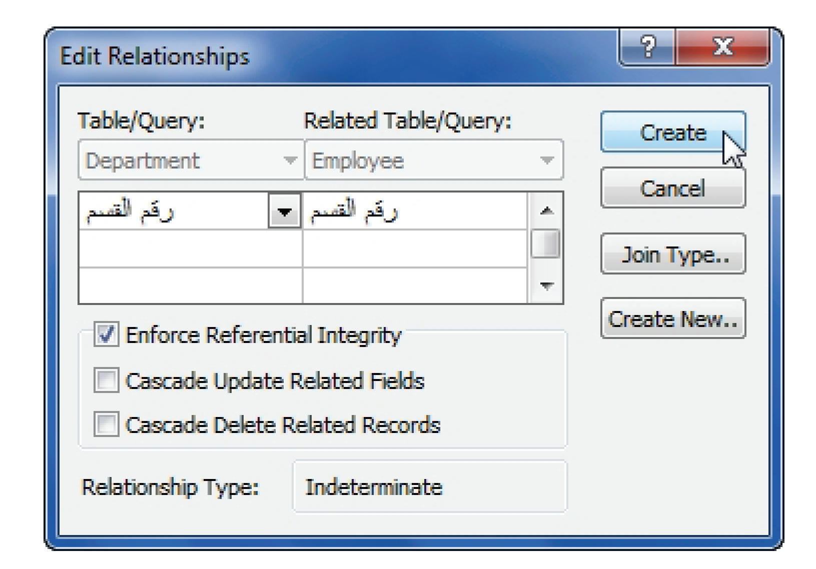
فتظهر النافذة التالية :



2 – حدد أسم الجدول الأول .. ثم إضغط مفتاح **Add** ، وحدد أسم الجدول الثانى ثم إضغط مفتاح **Add** مرة أخرى .. أو إضغط على أسم الجدول مرتين .. فتظهر الجداول .. كما بالشكل التالى :



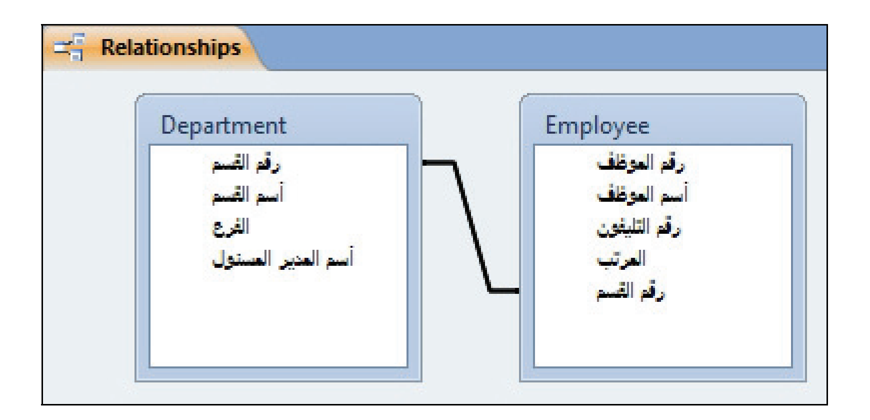
3 – أسحب بالـ Mouse حقل رقم القسم من جدول الأقسام Department وضع هذا الحقل على حقل رقم القسم فى جدول الموظفين Employee .. فتظهر النافذة Edit Relationships والتى توضح نوع العلاقات .. كما بالشكل التالى :



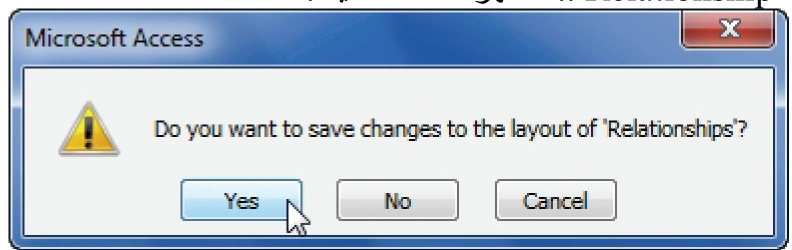


نوع العلاقة الموجودة بين الجدولين السابقين .. هى علاقة رأس بأطراف One to Many .

4 – إضغط مفتاح **Create** الموجود بالنافذة Edit Relationship السابقة .. فيظهر خط يربط حقول كل من الجدولين Department ، و Employee وهو حقل رقم القسم .. كما بالشكل التالى :



5 – إضغط على أداة الإغلاق Close الموجودة أعلى النافذة Relationship .. فتظهر النافذة التالية :



تسألك هل ترغب فى حفظ العلاقات السابقة .. إضغط **Yes** .. فيتم الحفظ والإغلاق .

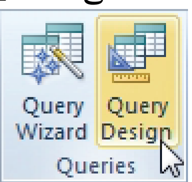


لكى تظهر نتيجة العلاقة بين الجدولين السابقين .. لابد من إنشاء إستعلامات أو إستفسارات Query .

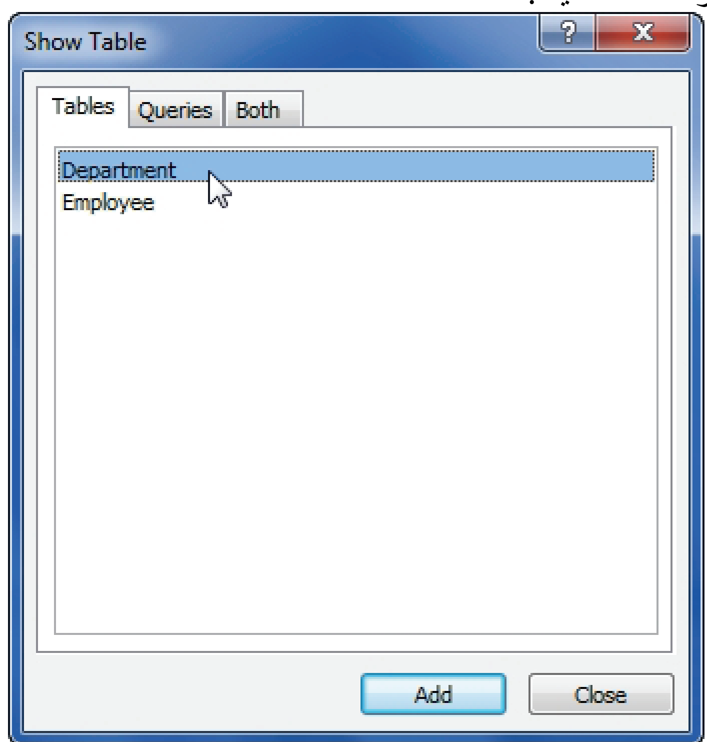
■ رابعاً : إنشاء الإستفسارات Quers :

لإنشاء إستفسارات بين الجداول .. إتبع الخطوات التالية :

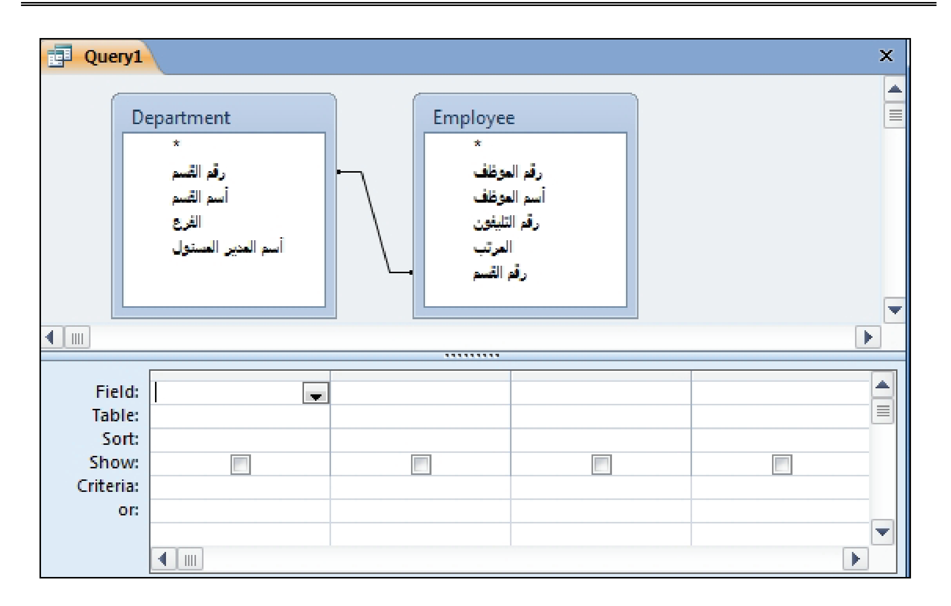
1 – من القائمة Create .. إضغط على الأداة Query Design ..



فتظهر النافذة التالية :



2 – حدد أسم الجدول .. ثم أضغط مفتاح **Add** ، أو إضغط مرتين على أسم الجدول فتظهر نافذة الإستفسارات .. كما بالشكل التالى :



3 – إختار حقول معينة من الجدولين السابقين .. وذلك بالضغط مرتين على أسم الحقل من كل جدول ، أو سحب الحقل بالـ Mouse ووضعه فى المستطيل Field الموجود بنافذة الإستعلامات .. فمثلاً :

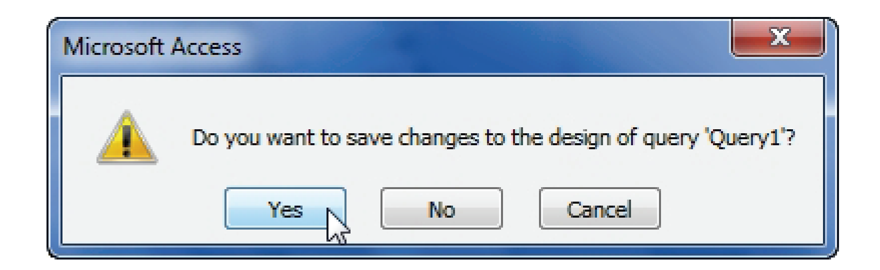
• إختار حقول (رقم الموظف / أسم الموظف / رقم التليفون / رقم القسم) من جدول الموظفين Employee .

• ثم إختار حقول (أسم القسم / الفرع) من جدول الأقسام Department .

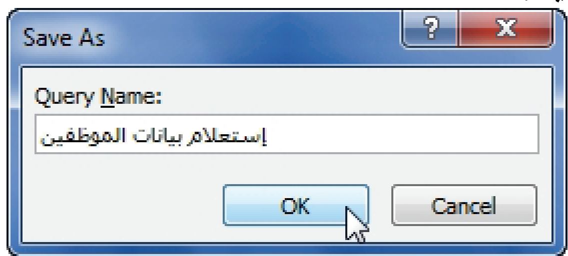
فيظهر هيكل الإستفسارات .. كما بالشكل التالى :

Field:	رقم الموظف	أسم الموظف	رقم التليفون	رقم القسم	أسم القسم	الفرع
Table:	Employee	Employee	Employee	Employee	Department	Department
Sort:						
Show:	☑	☑	☑	☑	☑	☑
Criteria:						
or:						

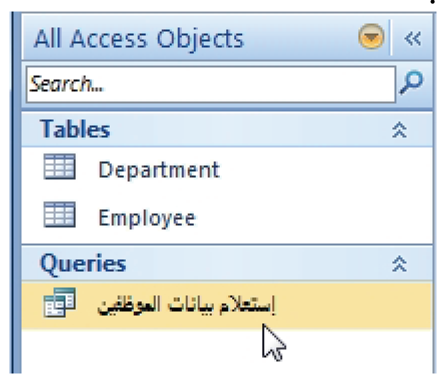
4 – إضغط على أداة الإغلاق × Close الموجودة أعلى النافذة Query.. فتظهر النافذة التالية :



تسألك هل ترغب فى حفظ الإستفسارات .. إضغط **Yes** للحفظ .. فتظهر النافذة التالية :



5 – إدخل إسم الإستعلام وهو .. (إستعلام بيانات الموظفين) .. ثم إضغط **OK** .. فيظهر أسم الإستعلام فى نافذة قاعدة البيانات .. كما بالشكل التالى :





لكى تظهر نتيجة الإستفسارات ، والعلاقات .. التى تم إنشاؤها فى الخطوات السابقة .. لابد من إدخال بيانات فى كل من جدول الموظفين Employee ، وجدول الأقسام Department .

■ خامساً : إدخال البيانات إلى الجداول :

1 – إضغط مرتين على جدول الأقسام Department .. فيتم فتحه .. ثم إدخل البيانات الخاصة بكل قسم .. كما بالشكل التالى :

رقم القسم	أسم القسم	الفرع	أسم المدير المسئول
1	نظم المعلومات	إسكندرية	ياسر عمر
2	التسويق	القاهرة	عادل سعيد
3	الإنتاج	بور سعيد	محمد البهنسى
4	الإدارة المالية	إسكندرية	رامى مروان
5	الموارد البشرية	القاهرة	سمير أبو الليل

• ثم قم بإغلاق الجدول بعد إدخال البيانات .. فيتم حفظ البيانات تلقائياً .

2 – إضغط مرتين على جدول الموظفين Employee .. فيتم فتحه .. ثم إدخل البيانات الخاصة بكل موظف .. كما بالشكل التالى :

رقم الموظف	أسم الموظف	رقم التليفون	المرتب	رقم القسم
1300	السيد أبو حفيظة	01000055661	$1,200.00	2
1301	وسيم هدهد	01112223334	$1,250.00	1
1305	وليد وحيد	01222233666	$1,500.00	3
1307	تامر سمير	01222244455	$2,000.00	5
1311	على عمر	01000111222	$1,700.00	4
1315	احمد شوقى	01112266999	$1,900.00	2
1317	نادر نور	01077788996	$2,000.00	1

■ سادساً : إظهار نتيجة العلاقات بين الجدولين السابقين :

إضغط مرتين على الإستفسار الموجود فى قاعدة البيانات بأسم (إستعلام بيانات الموظفين) .. فيتم فتحه .. فتظهر النافذة التالية :

رقم الموظف	أسم الموظف	رقم التليفون	رقم القسم	أسم القسم	الفرع
1301	وسيم هدهد	01112223334	1	نظم المعلومات	إسكندرية
1317	نادر نور	01077788996	1	نظم المعلومات	إسكندرية
1300	السيد أبو حفيظة	01000055661	2	التسويق	القاهرة
1315	احمد شوقى	01112266999	2	التسويق	القاهرة
1305	وليد وحيد	01222233666	3	الإنتاج	بور سعيد
1311	على عمر	01000111222	4	الإدارة المالية	إسكندرية
1307	تامر سمير	01222244455	5	الموارد البشرية	القاهرة

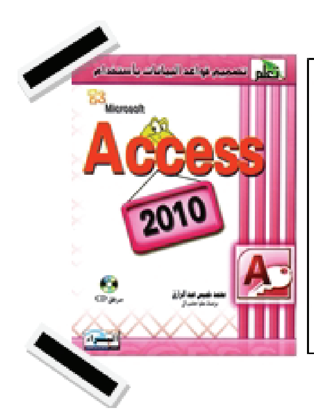


أنه ظهر الحقول التى تم إختيارها عند إنشاء الإستفسار.. وظهر أيضاً رقم القسم ، وأسم القسم تلقائياً .. وبالتالى أصبح هناك علاقة إرتباط بين الجدولين عند إدخال أى بيانات فى كل منهم .

- وبنفس الخطوات السابقة يمكنك إنشاء العديد من (الجداول ، والإستفسارات ، والعلاقات) داخل قاعدة البيانات .



لتعلم المزيد من تصميم قواعد البيانات .. يمكنك مراجعة كتاب (تصميم قواعد البيانات Access) ، أو مشاهدة الفيديو الخاص بتصميم قواعد البيانات الخاص بهذا الفصل على الموقع الخاص بى :
www.connect-eg.net
أو من خلال رابط الـ Facebook الخاص بدار البراء للنشر الموجود على هذا الموقع .



وبعد أن تعلمنا تصميم قواعد البيانات فى الخطوات السابقة .. نتعلم الآن كيفية تصميم واجهات الإستخدام وربط قاعدة البيانات بلغة الـ Visual Basic .

■ المرحلة الثانية : تصميم واجهات التطبيق وربطها بقاعدة البيانات :

1 – إفتح مشروع جديد New Project .. ثم غير الأسم الخاص بقاعدة البيانات ليصبح Data Base .. وحدد مكان قاعدة البيانات داخل المجلد الخاص بقواعد البيانات السابق إنشاؤه فى المرحلة الأولى .

2 – أضف عدد 5 نماذج Form للمشروع .. ثم غير الخاصية Text لكل نموذج .. بحيث تصبح .. كما يلى :
- Form 1 .. Home .
- Form 2 .. Employee .
- Form 3 .. Data Edit .
- Form 4 .. Department .
- Form 5 .. Show Query .

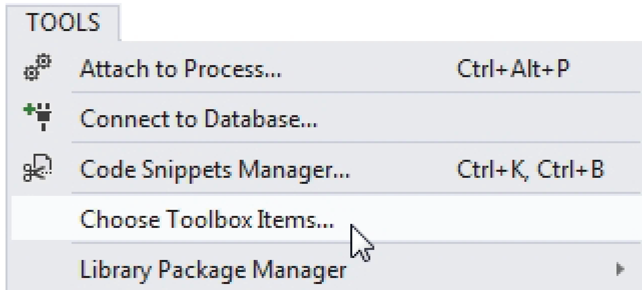
3 – غير لون الخلفية لكل Form .. من الخاصية Back Color باللون المناسب .

4 – غير الخاصية Size ، والخاصية Maximum Size لكل نموذج ليصبح :

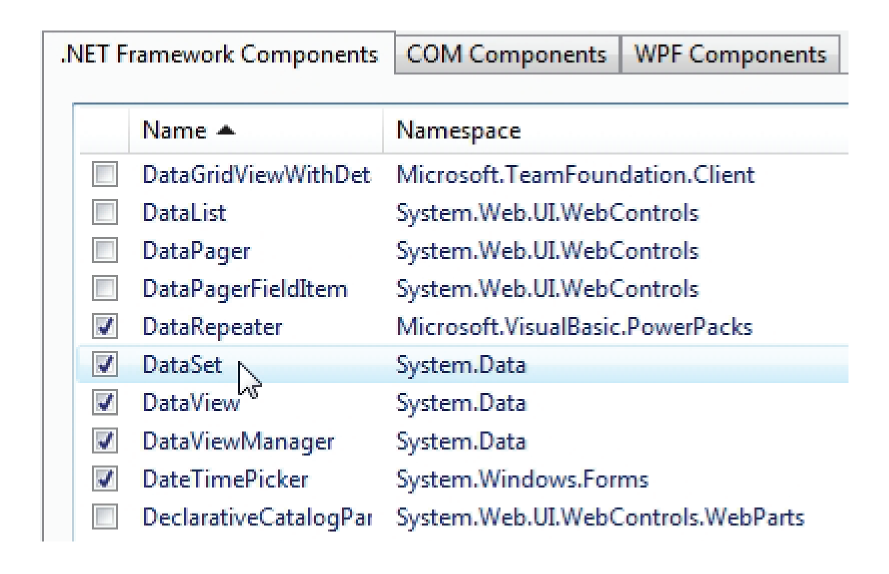
- Width 500 .
- Height 500 .

⊟ MaximumSize	**500, 500**
Width	**500**
Height	**500**
⊞ MinimumSize	0, 0
⊞ Padding	0, 0, 0, 0
⊞ Size	**500, 500**

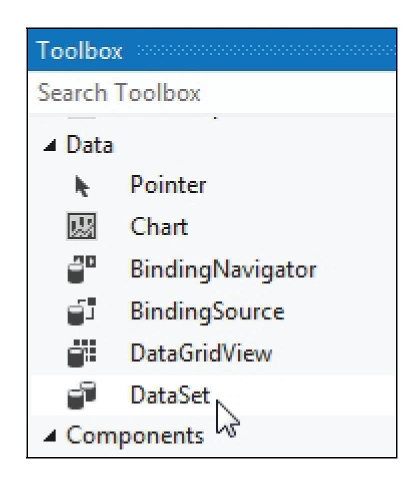
5 – قف على النموذج Form 2 .. الخاص ببيانات الموظفين .. ثم من القائمة Tools إختار الأمر Choose Tool Box Items ..



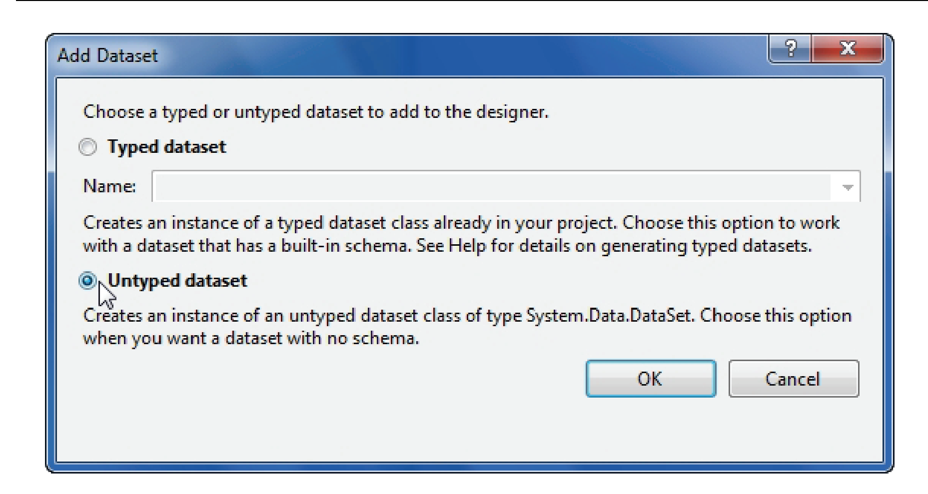
فتظهر النافذة التالية :



6 – نشط الإختيار Data Set .. ثم إضغط **OK** .. فتظهر الأداة Data Set في صندوق الأدوات Tool Box .. كما بالشكل التالى :



7 – إضغط مرتين على الأداة Data Set .. فتظهر النافذة التالية :



8 – نشط الإختيار Un Typed Data Set .. ثم إضغط **OK** .. فتظهر الأداة Data Set أسفل نافذة النموذج Form 2 .

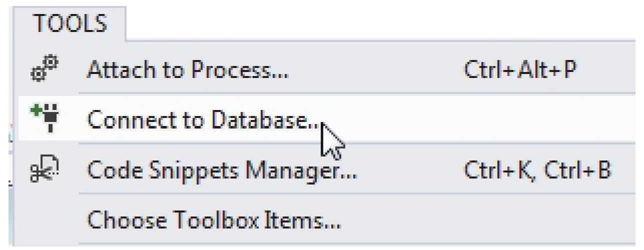
9 – أضف عدد 5 Text Box .. والتى تمثل حقول جدول بيانات الموظفين الموجود بملف قاعدة البيانات .

10 – أضف عناوين Label للنموذج Form 2 بالعدد المناسب لواجهة الإستخدام .. ثم غير الخاصية Text لتناسب كل عنوان .

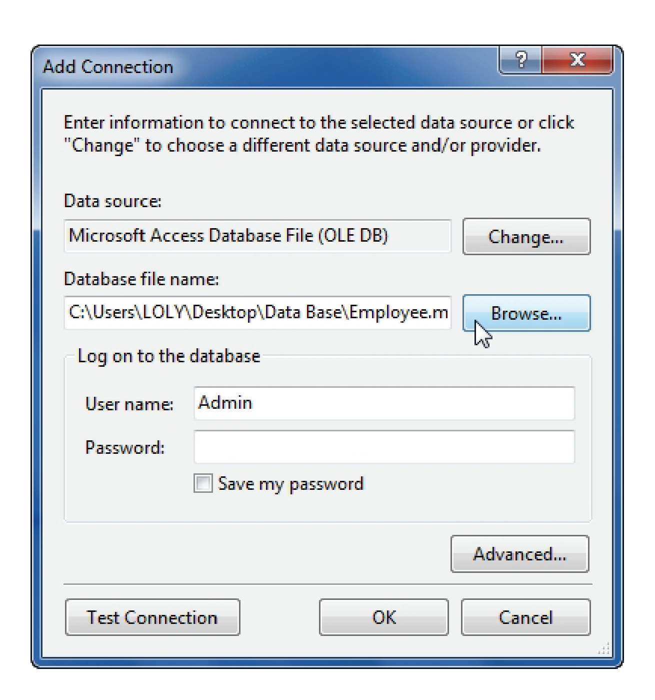
11 – أضف عدد 3 مفاتيح تحكم Button للنموذج Form 2 .. ثم غير الخاصية Text لكل مفتاح .. ليصبح كل مفتاح (الرئيسية / التالى / السابق / إغلاق) .. فيظهر النموذج Form 2 .. كما بالشكل التالى :



12 – قم بعمل إتصال بقاعدة البيانات .. من القائمة Tools إختار .. .. Connect Data Base

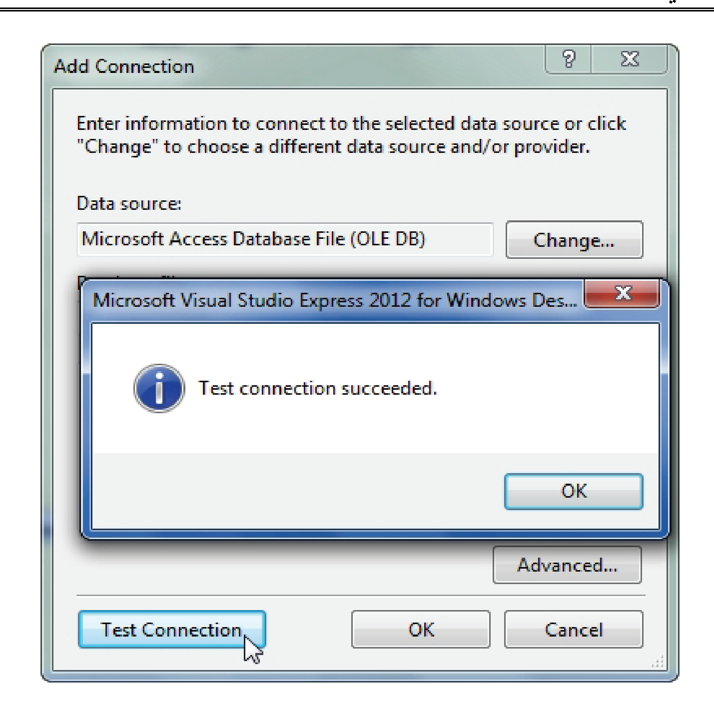


فتظهر النافذة التالية :

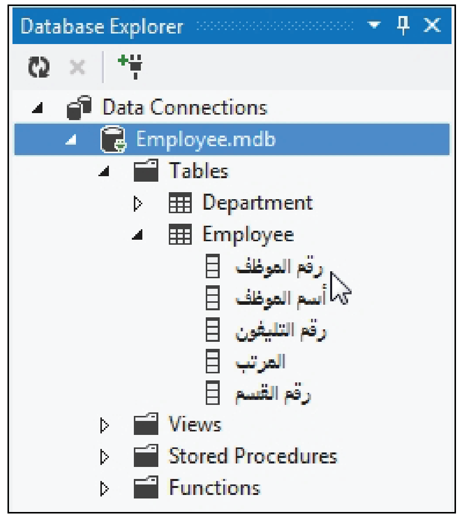


13 – إضغط Browse لتحديد مكان ملف قاعدة البيانات Access .. ثم إضغط Open للعودة للنافذة السابقة Add Connection .

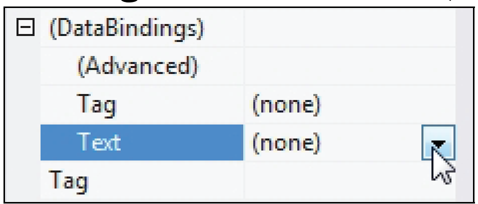
14 – إضغط مفتاح Test Connection .. وذلك إختبار صحة الإتصال بقاعدة البيانات .. فتظهر رسالة عملية الإتصال بنجاح .. كما بالشكل التالى :



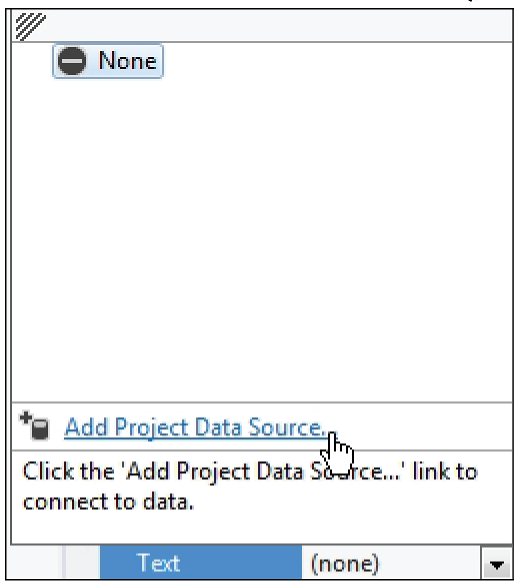
15 – إضغط **OK** .. ثم إضغط **OK** مرة أخرى .. فتظهر النافذة .. Data Base Explorer يسار نافذة البرنامج .. وبها أسماء جداول ، وحقول قاعدة البيانات .. كما بالشكل التالى :



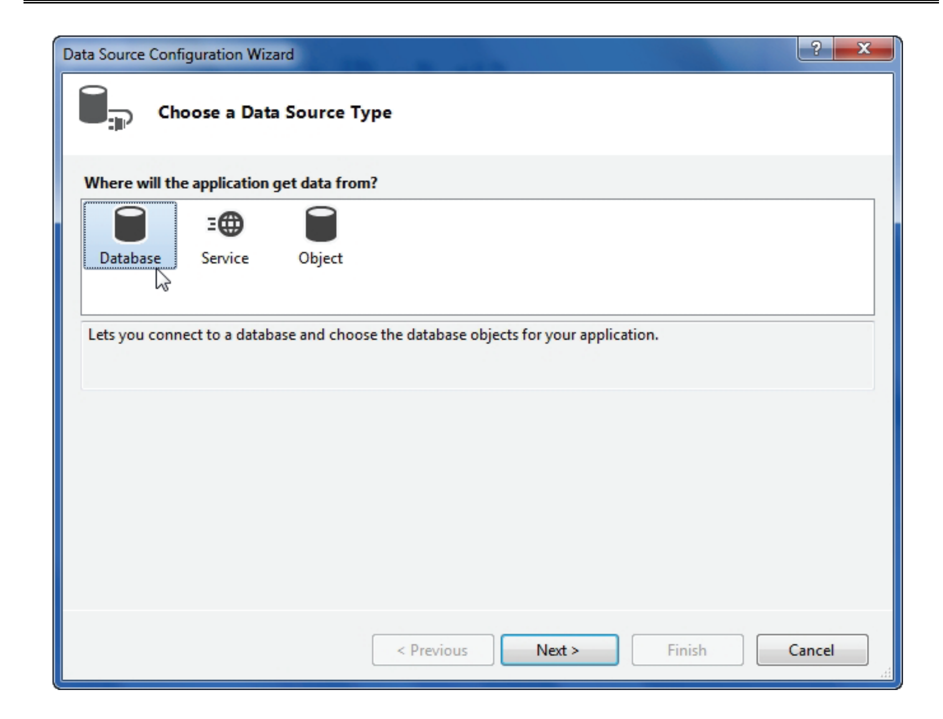
15 – قم بربط كل حقل داخل النموذج وهو Text Box .. بالحقل المناظر له فى ملف قاعدة البيانات Access .. وذلك بتحديد على أول Text Box ثم من الخاصية Data Bindings .. إضغط على المفتاح الصغير بجانب الخاصية Text .. كما بالشكل التالى :



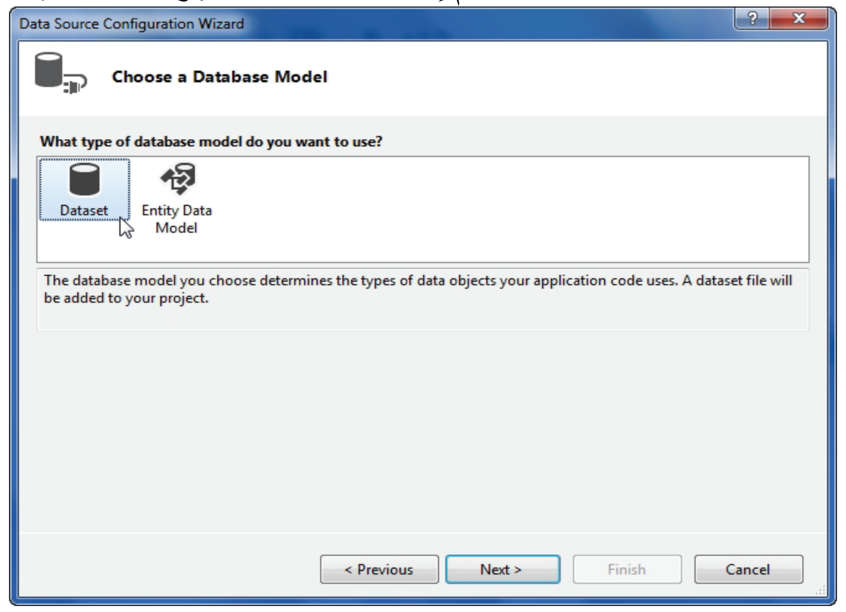
فتظهر القائمة التالية :



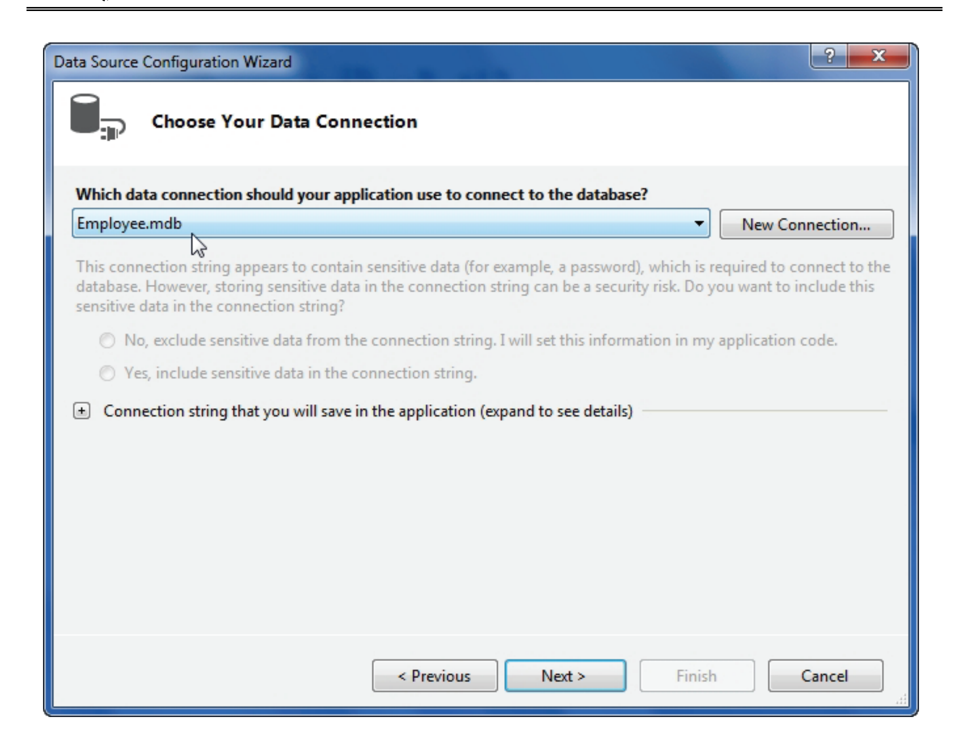
16 – إضغط على الأمر Add Project Data Source .. فتظهر النافذة التالية :



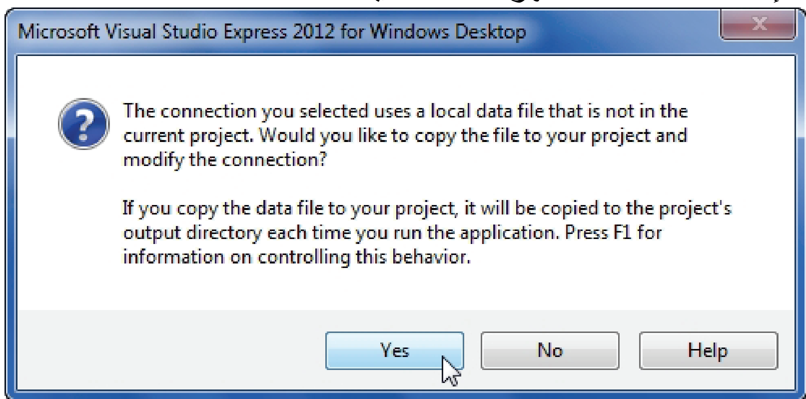
17 – حدد الأداة Data Base .. ثم إضغط **Next** .. فتظهر النافذة التالية :



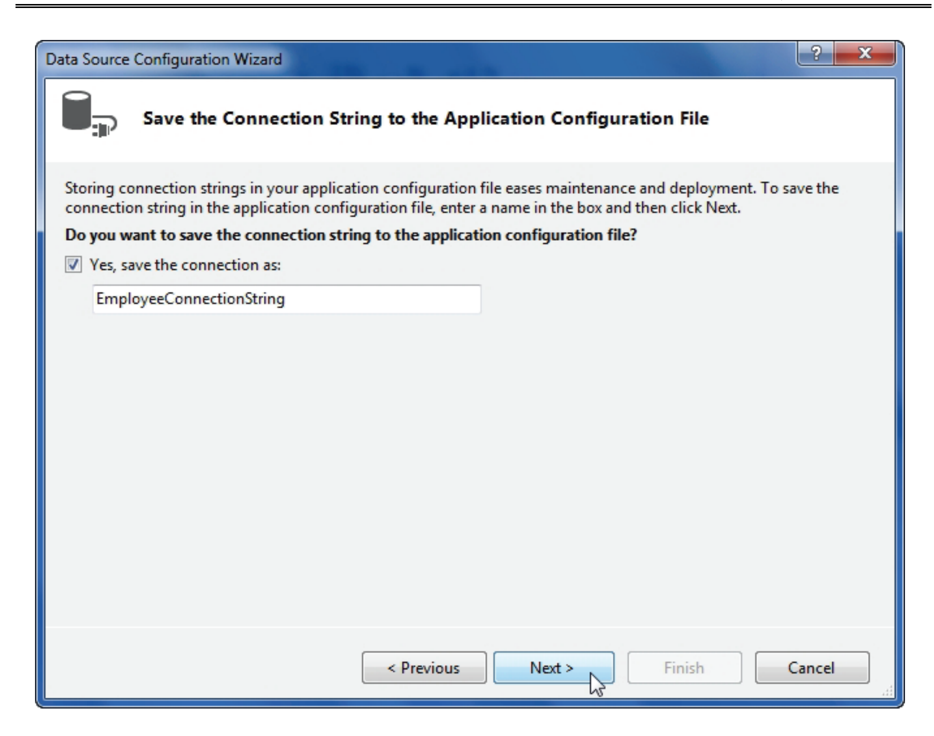
18 – حدد الأداة Data Set .. ثم إضغط **Next** .. فتظهر النافذة التالية :



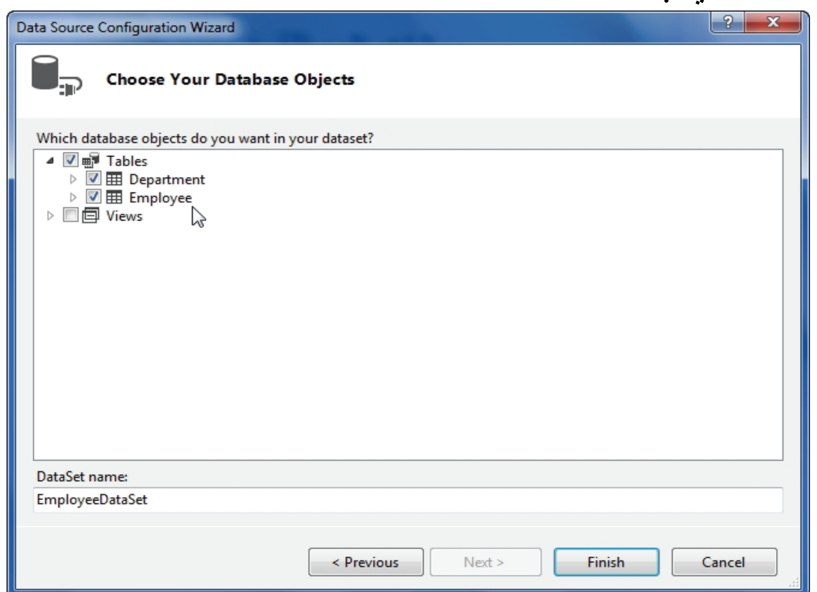
19 – حدد مكان ملف قاعدة البيانات Access .. إن لم يكن ظاهراً .. ثم إضغط **Next** .. فتظهر النافذة التالية :



20 – إضغط **Yes** .. فتظهر النافذة التالية :



يظهر بها أسم ملف قاعدة البيانات Access .. ثم إضغط **Next** .. فتظهر النافذة التالية :



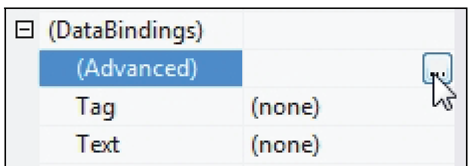
وبها أسماء جداول ، وحقول قاعدة البيانات التى تم الإتصال بها .. ثم إضغط **Finish** .. للإنتهاء من الإتصال بقاعدة البيانات .



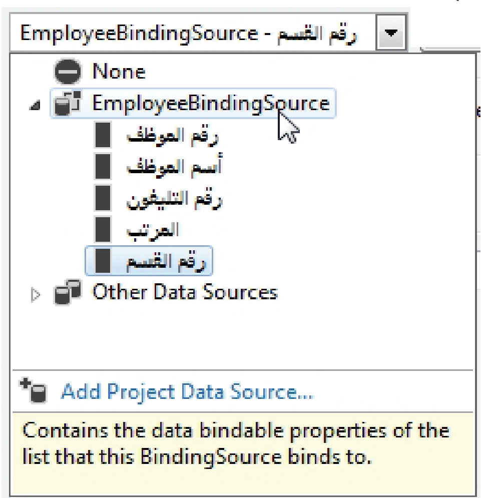
خطوات الإتصال السابقة هى الإتصال بقاعدة البيانات ككل .. ولكن لم نتصل حتى الأن بملف الموظفين Employee داخل قاعدة البيانات .. لتحديد كل حقل .

■ الإتصال بحقول ملف قاعدة البيانات Employee :

1 – حدد الـ Text Box الخاص بحقل رقم الموظف .. ثم من الخاصية .. Data Binding .. إضغط على المفتاح الصغير الموجود بجانب الخاصية Advanced ..



فتظهر النافذة التالية :



2 – من القائمة المنسدلة Binding .. إختار Project Data Source .. فتظهر قاعدة البيانات بأسم Employee Data Set .. ومنها ملف الموظفين Employee .. وبالضغط عليه تظهر أسماء حقول الجدول .

3 – إختار الحقل الأول .. وهو رقم الموظف .. ثم إضغط **OK** .
4 – كرر الـ 3 خطوات السابقة لباقى حقول جدول قاعدة البيانات .

■ كتابة أكواد مفاتيح التحكم الخاصة بالنموذج Form 2 :

• **كود مفتاح " الرئيسية " :**

```
Form1.show
```

• **كود مفتاح " السابق " :**

```
Me.EmployeeBindingSource.MoveLast
```

• **كود مفتاح " التالى " :**

```
Me.EmployeeBindingSource.MoveNext
```

• **كود مفتاح " إغلاق" :**

```
End
```

■ نموذج إضافة وتعديل بيانات الموظفين Form 3 :

نفس شكل النموذج Form 2 .. ولكن مع إختلاف شكل مفاتيح التحكم .



يمكن نسخ عناصر وأدوات التحكم الخاصة بالـ Form 2 .. بدلاً من إعادة تصميمها مرة أخرى .

ولتنفيذ شكل النموذج Form 3 .. إتبع الخطوات التالية :

1 – أضف الأداة Binding Source من صندوق الأدوات .. وهى الخاصة بالإتصال بقاعدة البيانات .. السابق تنفيذها فى الخطوات السابقة .

2 – إربط كل حقل بالنموذج Form 3 .. بالحقل المناظر له فى قاعدة البيانات Access .. كما تعلمت فى النموذج Form .

■ كتابة أكواد مفاتيح التحكم الخاصة بالنموذج Form 3 :

• **كود مفتاح " جديد " :**

```
Me.EmployeeBindingSource.AddNew()
```

• **كود مفتاح " حفظ" :**

```
Me.EmployeeBindingSource.EndEdit
```

• **كود مفتاح "حذف" :**

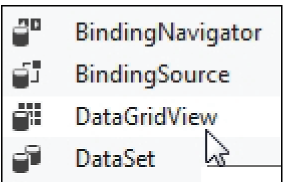
```
Me.EmployeeBindingSource.RemoveCurrent
```

• **كود مفتاح "إغلاق" :**

```
End
```

■ نموذج بيانات الأقسام Form 4 :

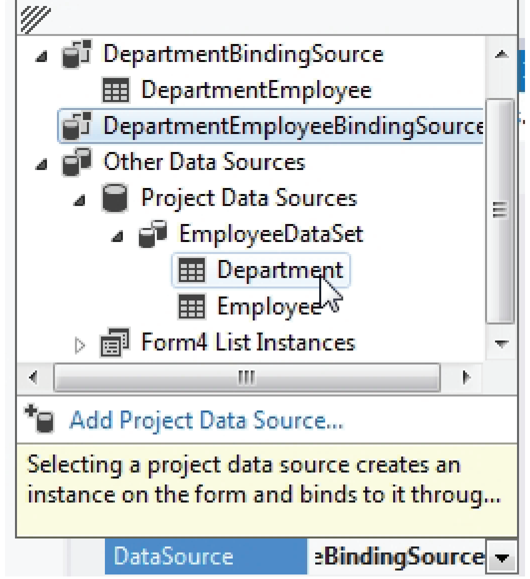
1 – إفتح النموذج Form 4 .. ثم أضف الأداة Data Grid Viwe من صندوق الأدوات Tool Box ..



2 – أضف مفتاح للتحكم Button .. بأسم " إغلاق " .. ثم أكتب الكود التالى للمفتاح " إغلاق " :

End

3 – أربط الأداة Data Grid Viwe الموجودة فى النافذة Form 4 بجدول الأقسام .. وذلك بإختيار الخاصية Data Source .. ثم إختار أسم الجدول Department Employee Binding Source .. ثم Department كما بالشكل التالى :



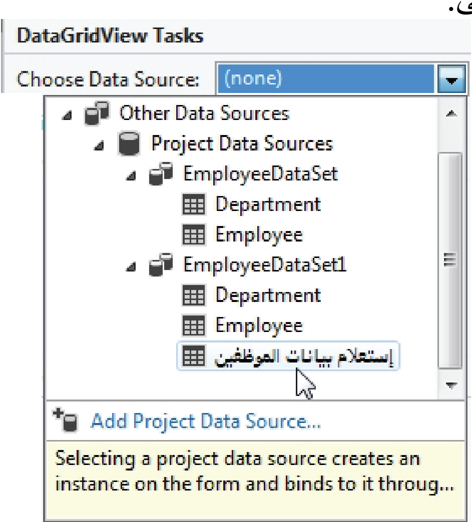
فتظهر نافذة النموذج كما بالشكل التالى :



■ نموذج عرض إستعلامات بيانات الموظفين Form 5 :

ولتنفيذ تطبيق النموذج Form 5 .. إتبع الخطوات التالية :

1 – نفس خطوات النموذج السابق Form 4 .. ولكن مع ربط النافذة .. Data Grid Viwe .. بالإستعلام .. (إستعلام بيانات الموظفين) .. كما بالشكل التالى:



2 – أضف مفتاح Button .. بأسم إغلاق .. وأكتب له الكود التالى :

```
End
```

فتظهر نافذة النموذج .. كما بالشكل التالى :



وأخيراً .. يتبقى لنا فى هذه المرحلة .. عملية ربط مفاتيح التحكم الخاصة بالواجهة الرئيسية النموذج Form 1 .

■ نموذج الواجهة الرئيسية للبرنامج Form 1 :

1 – إضف عنوان Label ، ثم أضف الصورة الخاصة بالشركة عن طريق الأداة Picture Box .. كما تعلمت سابقاً .. ثم أضف عدد 4 مفاتيح تحكم Button .. وهى كالتالى (بيانات الموظفين / إضافة وتعديل البيانات / بيانات الأقسام / عرض بيانات الموظفين) .. فتظهر نافذة نموذج الواجهة الرئيسية .. كنا بالشكل التالى :



2 – أكتب الأكواد التالية للمفاتيح الأربعة السابقة :

- **كود مفتاح "بيانات الموظفين" :**

```
Form2.Show
```

- **كود مفتاح "إضافة وتعديل البيانات" :**

```
Form3.Show
```

- **كود مفتاح "بيانات الأقسام" :**

```
Form4.Show
```

- **كود مفتاح "عرض بيانات الموظفين" :**

```
Form5.Show
```

3 – شغل البرنامج .. بالضغط على مفتاح **F5** .

■ المرحلة الثالثة : إخراج البرنامج للعمل من خلال أيقونة Setup :

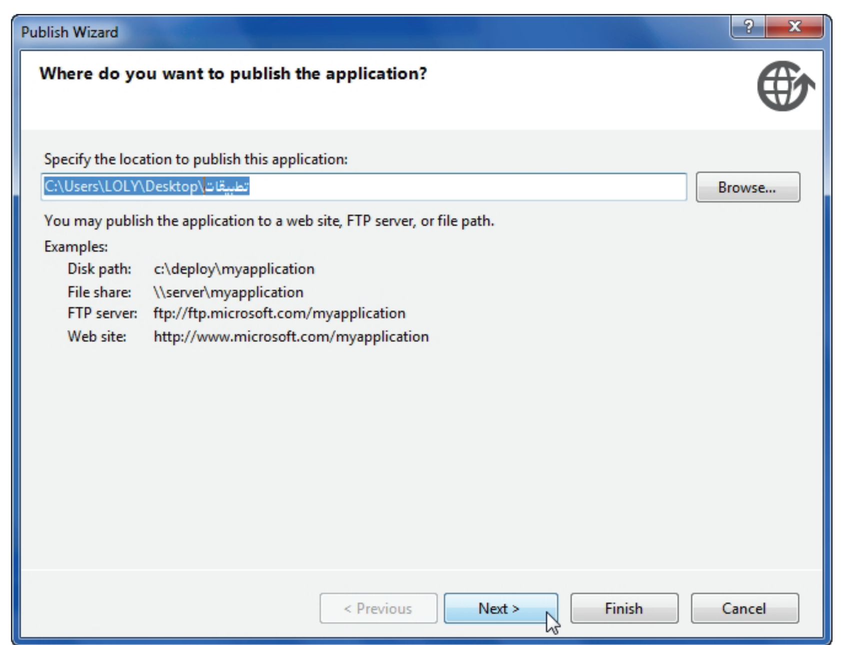
نقوم فى هذه المرحلة بإخراج البرنامج فى صورته النهائية للعمل على أى جهاز كمبيوتر .. لا يوجد به نسخة من برنامج Mocrosoft Visual Studio وذلك من خلال أيقونة الإعدادات Setup .. ولتنفيذ ذلك .. إتبع الخطوات التالية :

1 – من القائمة Build .. إختار الأمر Publish Windows Application

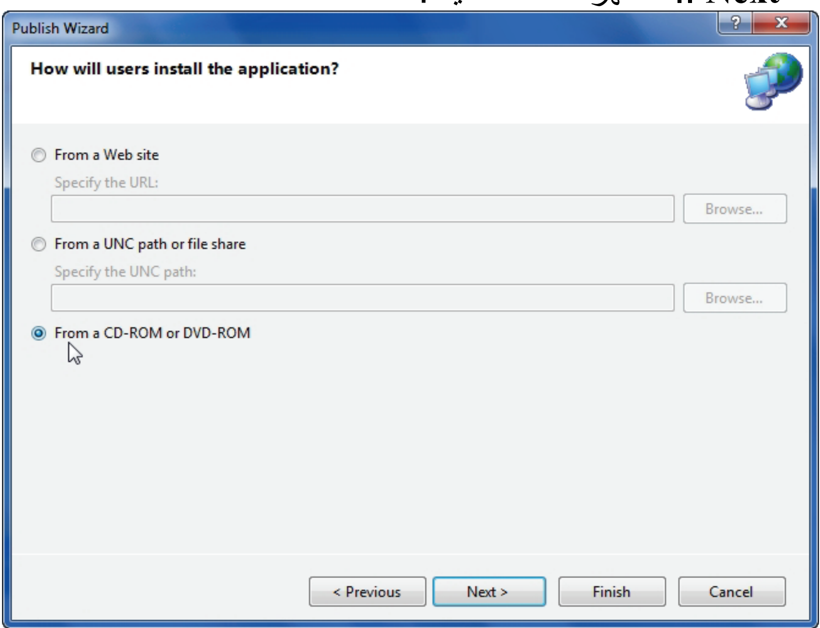
BUILD

- Build Solution F7
- Rebuild Solution Ctrl+Alt+F7
- Clean Solution
- Run Code Analysis on Solution Alt+F11
- Publish WindowsApplication1
- Run Code Analysis on WindowsApplication1
- Configuration Manager...

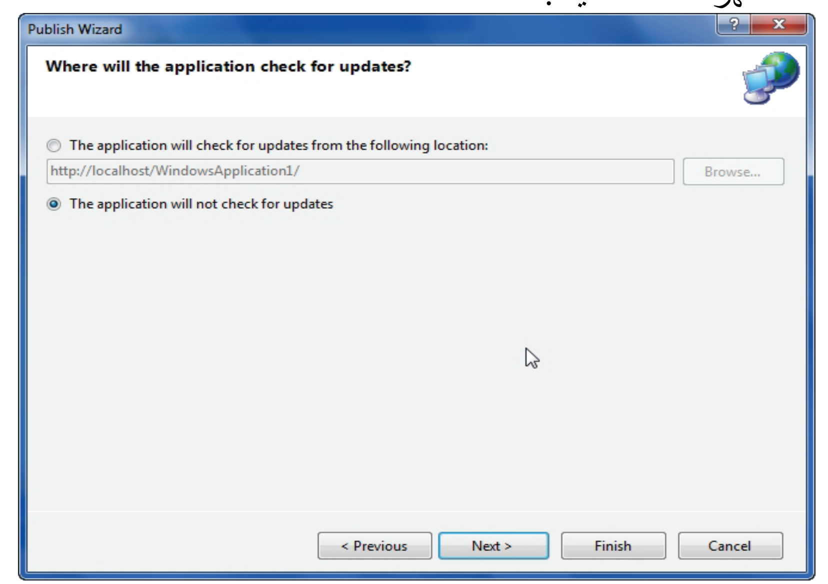
فتظهر النافذة التالية :



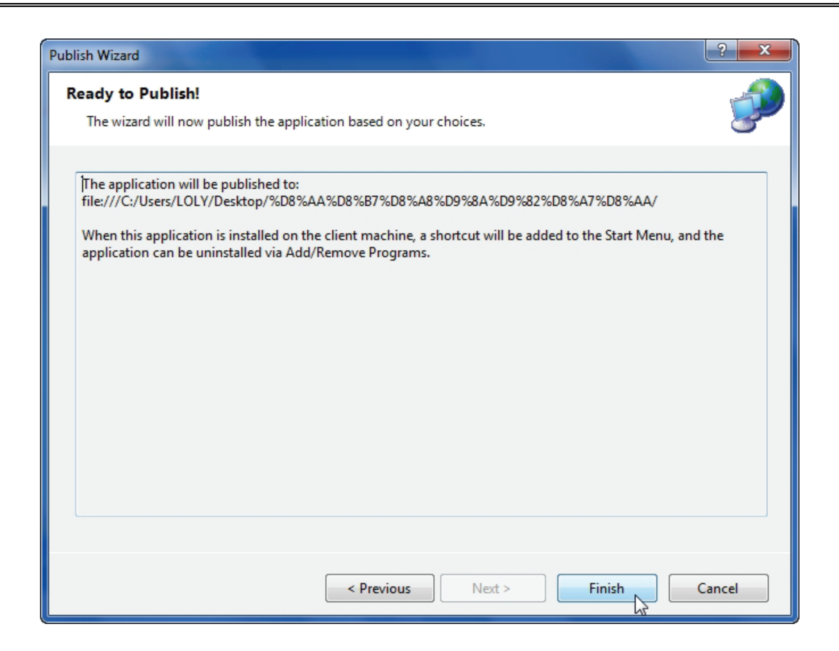
2 – حدد مكان حفظ البرنامج بالضغط على المفتاح **Browse** .. ثم إضغط **Next** .. فتظهر النافذة التالية :



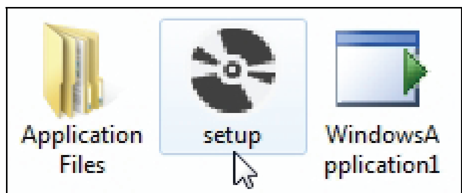
3 – نشط الإختيار Form a CD – Rom or DVD Rom .. وذلك لنقل البرنامج على أسطوانات CD ، أو DVD .. ثم إضغط Next .. فتظهر النافذة التالية :



4 – نشط الإختيار The Application Will not Check for Update .. ثم إضغط Next .. فتظهر النافذة التالية :

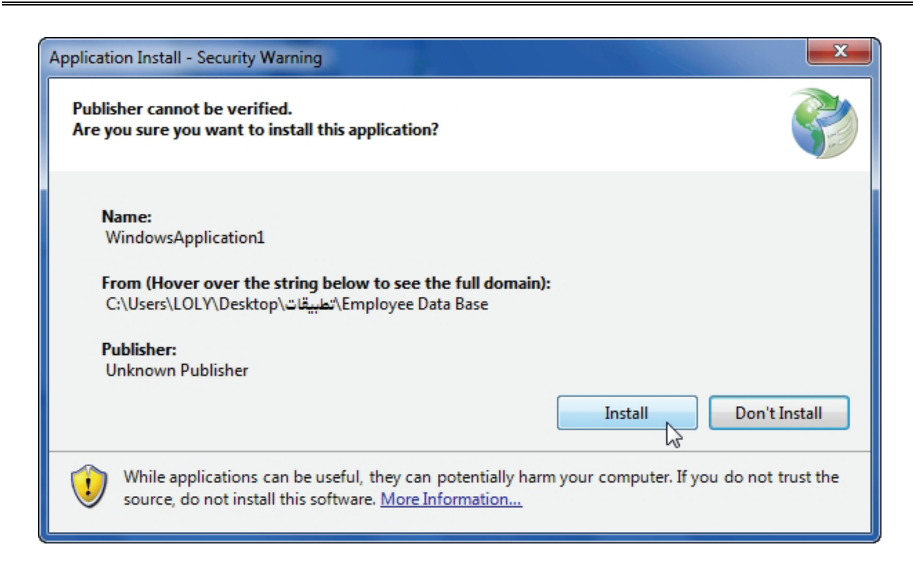


5 – إضغط **Finish** .. لإنهاء إعداد البرنامج للعرض .. فتظهر أيقونات البرنامج داخل المجلد Folder الخاص بقواعد البيانات .. كما بالشكل التالى :



■ تشغيل البرنامج للعرض :

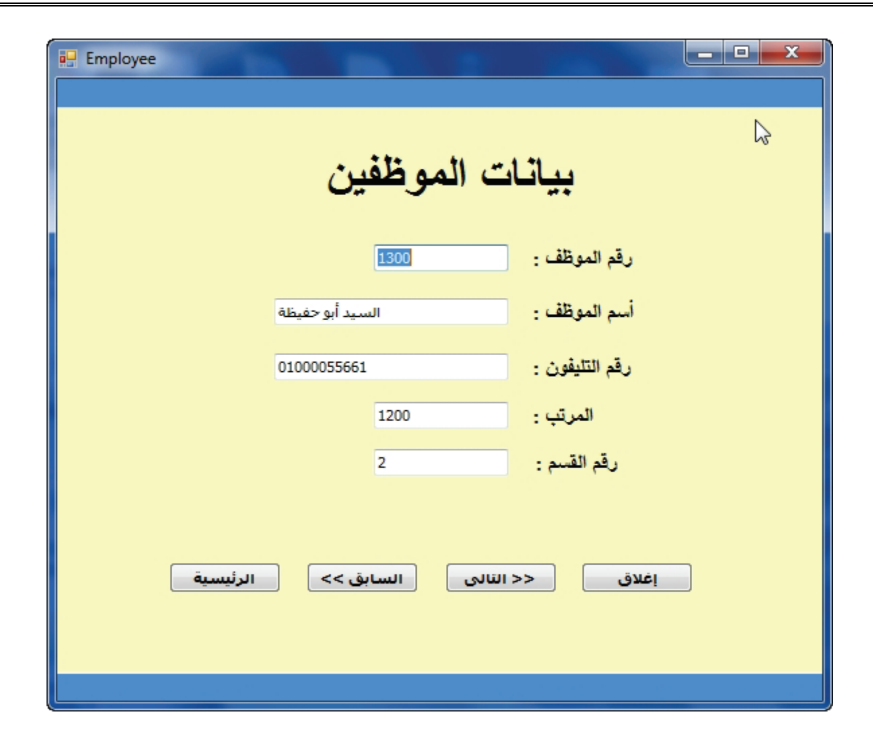
1 – إضغط مرتين على أيقونة Setup .. فتظهر النافذة التالية :



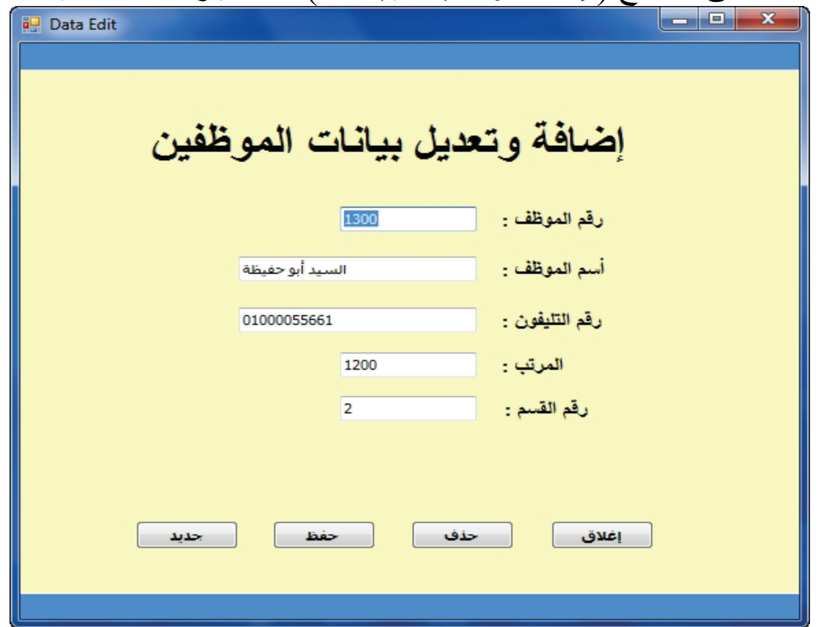
2 – إضغط **Istall** .. فيتم إعداد البرنامج على جهاز الكمبيوتر .. فتظهر النافذة الرئيسية للبرنامج .. كما بالشكل التالى :



• إضغط على المفتاح (**بيانات الموظفين**) .. فتظهر النافذة التالية :



• إضغط على المفتاح (**إضافة وتعديل البيانات**) .. فتظهر النافذة التالية :



• إضغط على المفتاح (**بيانات الأقسام**) .. فتظهر النافذة التالية :



رقم القسم	أسم القسم	الفرع	أسم المدير المسئول
1	نظم المعلومات	إسكندرية	ياسر عمر
2	التسويق	القاهرة	عادل سعيد
3	الإنتاج	بور سعيد	محمد البهنسى
4	الإدارة المالية	إسكندرية	رامى مروان
5	الموارد البشرية	القاهرة	سمير أبو الليل

• إضغط على المفتاح (**إستعلام بيانات الموظفين**) .. فتظهر النافذة التالية :



رقم الموظف	أسم الموظف	رقم التليفون	رقم القسم	أسم القسم
1301	وسيم هدهد	01112223334	1	نظم المعلومات
1317	نادر نور	01077788996	1	نظم المعلومات
1300	السيد أبو حفيظة	01000055661	2	التسويق
1315	احمد شوقى	01112266999	2	التسويق
1305	وليد وحيد	01222233666	3	الإنتاج
1311	على عمر	01000111222	4	الإدارة المالية
1307	تامر سمير	01222244455	5	الموارد البشرية

■ **وبذلك :**

يمكنك أن تقوم بنفسك بتصميم العديد من قواعد البيانات ، وربطها بلغة الـ Visual Basic .. وتصميم برامج متكاملة .. إلا أنه هناك العديد من الطرق والأفكار لتصميم وإدارة قواعد البيانات .

الفصل الثانى عشر
تطبيقات بإستخدام لغة
الـ Visual Basic

نتعلم فى هذا الفصل ..



تصميم التطبيقات المختلفة بلغة الـ Visual Basic :

- تصميم متصفح إنترنت Web Browser .
- تصميم آله حاسبه Calculator .
- تصميم برنامج لحساب زكاة المال .
- تصميم نافذة إفتتاحية لواجهة البرامج Splash Screen .
- تصميم نافذة لإظهار كلمة سر للبرنامج ..
Password Login Form
- تصميم شريط القوائم داخل البرامج Menu Bar .
- تصممصم نص متحرك داخل نافذة البرنامج .

الفصل الثانى عشر

تطبيقات باستخدام لغة الـ Visual Basic

■ تطبيق 1 : تصميم متصفح إنترنت Web Browser

لتصميم متصفح إنترنت يشبه Internet Explorer ، أو Fire Fox ..
إتبع الخطوات التالية :

■ أولاً : تصميم واجهة المتصفح :

1 – إفتح مشروع جديد New Project .

2 – حدد أسم المشروع ، ومكان الحفظ .. من النافذة New Project .. كما بالشكل التالى :

Name:	Web Browser
Location:	C:\Users\LOLY\Desktop\تطبيقات\
Solution:	Create new solution
Solution name:	Web Browser

فتظهر نافذة وواجهة العمل على بيئة الـ Visual Basic .

3 – غير أسم النافذة Form 1 من الخاصية Text .. لتصبح Web Browser .

4 – غير الخاصية Maximum Size .. ليصبح الـ Width 1280 ، والـ Height 1024 .. وهى أقصى درجة لتكبير المتصفح على شاشة الكمبيوتر .

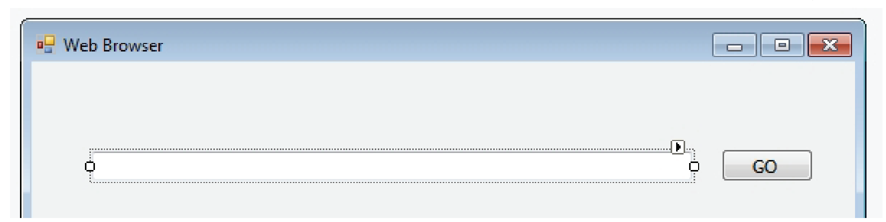
⊟ MaximumSize	1280, 1024
Width	1280
Height	1024

5 – غير الخاصية Size .. ليصبح الـ Width 1172 ، والـ Height 884 وهو المقياس الإفتراضى عند بداية فتح المتصفح على شاشة الكمبيوتر .

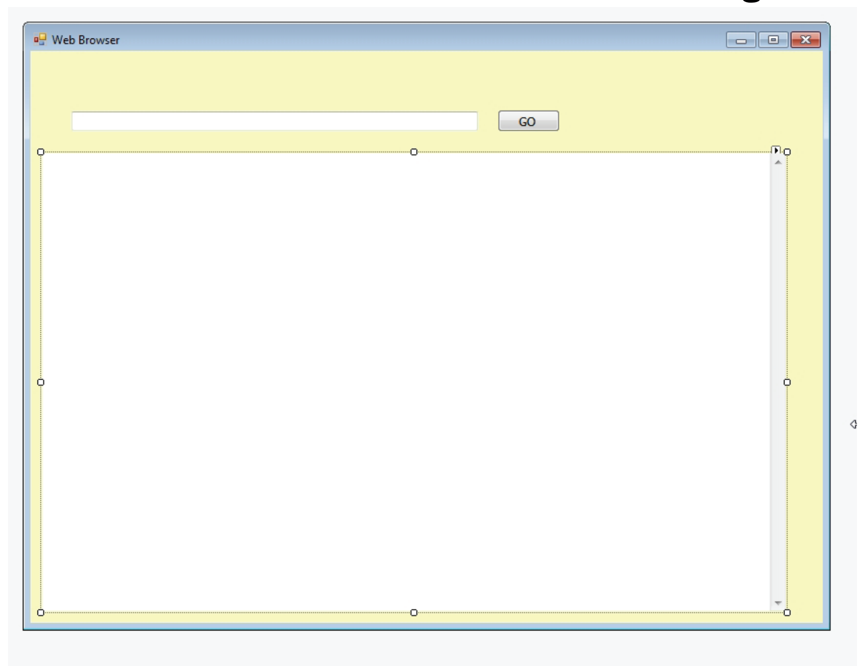
6 – غير لـون خلفية النموذج Form باللون المناسب من الخاصية Back Color .. وليكن اللون الأصفر .

7 – أضف Text Box.. ثم أضف مفتاح للتحكم Button للنافذة Form .

8 – غير الخاصية Text للمفتاح Button .. لتصبح **GO** .. كما بالشكل التالى :



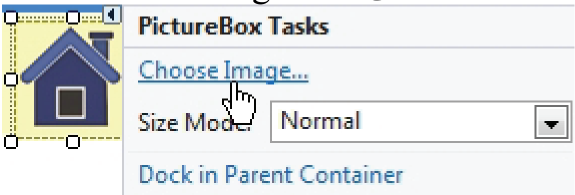
9 – أضف الأداة Web Browser [WebBrowser] من صندوق الأدوات Tool Box .. بالحجم المناسب داخل النافذة Form .. كما بالشكل التالى :



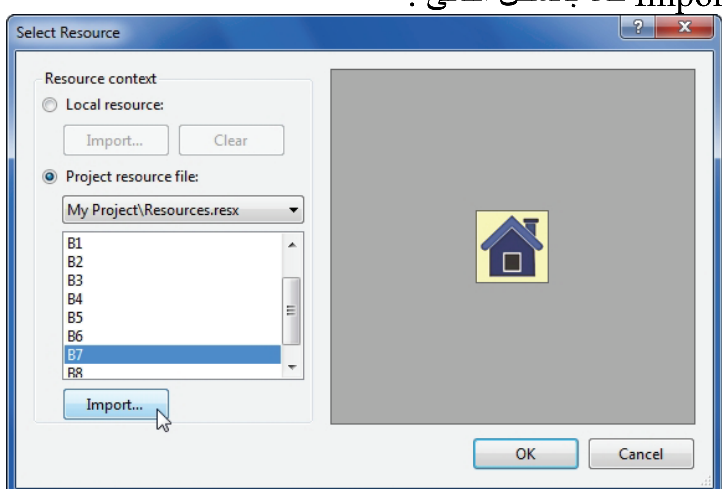
10 – أضف مستطيل Rectangle Shape من صندوق الأدوات .. ثم غير لون المستطيل من الخاصية Fill Color .. وضع هذا المستطيل أعلى نافذة النموذج Form .. ثم إنسخ هذا المستطيل وضعه أسفل نافذة النموذج .

11 – أضف الأداة Picture Box من صندوق الأدوات .. وذلك لوضع عدد 6 صور لتعبر عن أيقونات (Back / Forward / Home / Refresh / Stop / Option) .

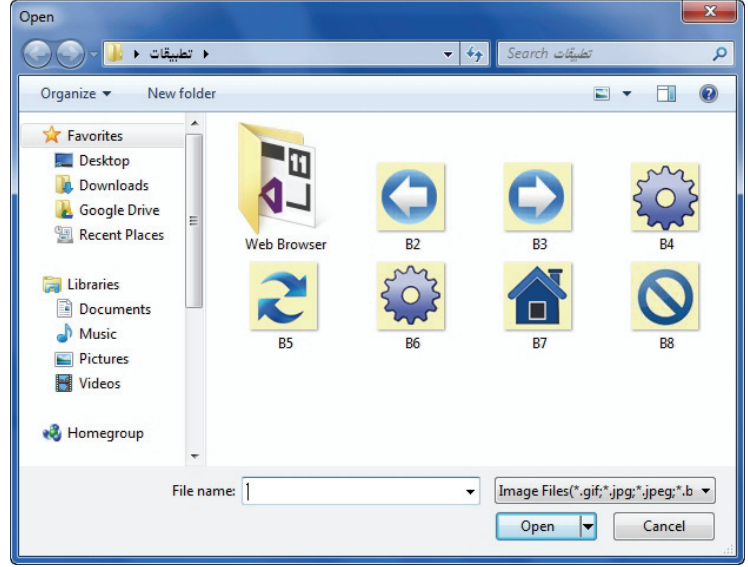
12 – إضغط على المثلث الصغير الموجود أعلى الأداة Picture Box فتظهر قائمة إضغط على الأمر Choose Image ..



فتظهر نافذة إختيار الصور Select Resource .. ثم إضغط مفتاح Import كما بالشكل التالى :

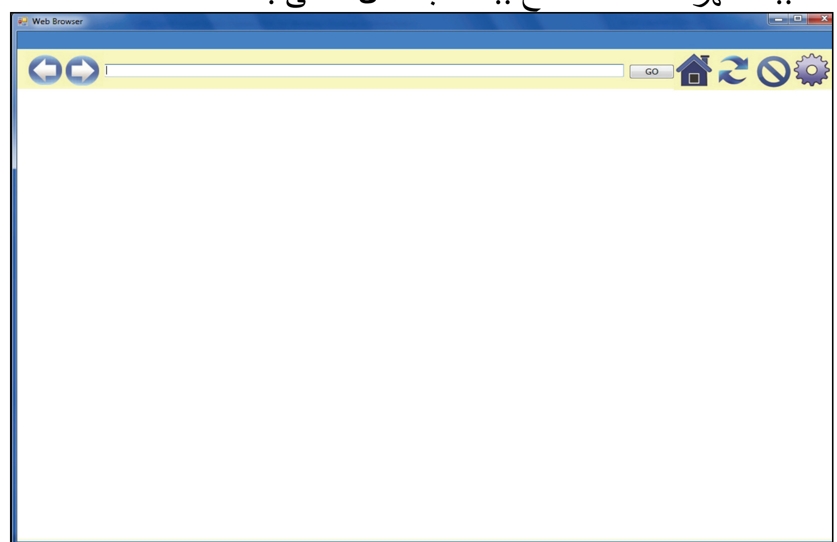


تظهر نافذة مكان إختيار الصور .. إختار الصور المناسبة لكل أداة .. كما بالشكل التالى :



14 – العودة إلى النافذة Select Resource .. ثم إضغط **OK** .. فتظهر الصورة فى شكل أيقونة داخل نافذة النموذج .

15 – كرر الـ 3 خطوات السابقة لعدد الأيقونات الموجودة فى نافذة المتصفح .. فتظهر نافذة المتصفح .. كما بالشكل التالى :



■ ثانياً : كتابة الأكواد البرمجية :



1 – إضغط مرتين على المفتاح GO مرتين .. فتظهر نافذة الكود .. وأكتب الكود التالى :

```
WebBrowser1.Navigate(TextBox1.Text)
```

2 – إضغط مرتين على المفتاح Back ثم أكتب الكود التالى :

```
WebBrowser1.GoBack
```

3 – كود المفتاح Forward :

```
WebBrowser1.GoForward
```

4 – كود المفتاح Home :

```
WebBrowser1.Navigate("www.google.com")
```

5 – كود المفتاح Refresh :

```
WebBrowser1.Refresh
```

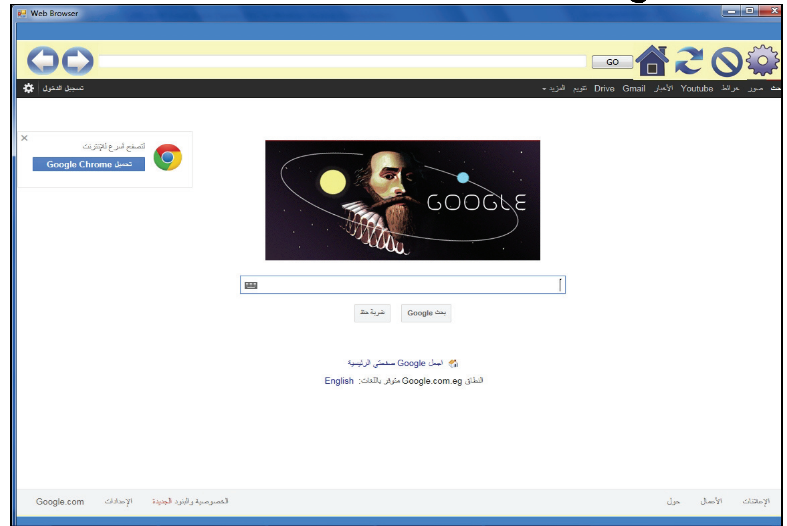
6 – كود المفتاح Stop :

```
WebBrowser1.Stop
```

7 – كود المفتاح Option :

```
Process.Start("inetcpl.cpl")
```

8 – شغل البرنامج .. بالضغط على **F5** .. فيظهر كما بالشكل التالى :

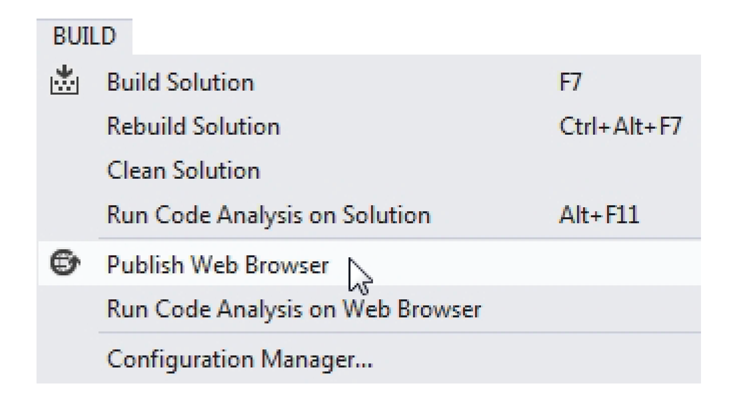


9 – إدخل على أى موقع تريده .. مثل www.yahoo.com .. فيظهر المتصفح كما بالشكل التالى :

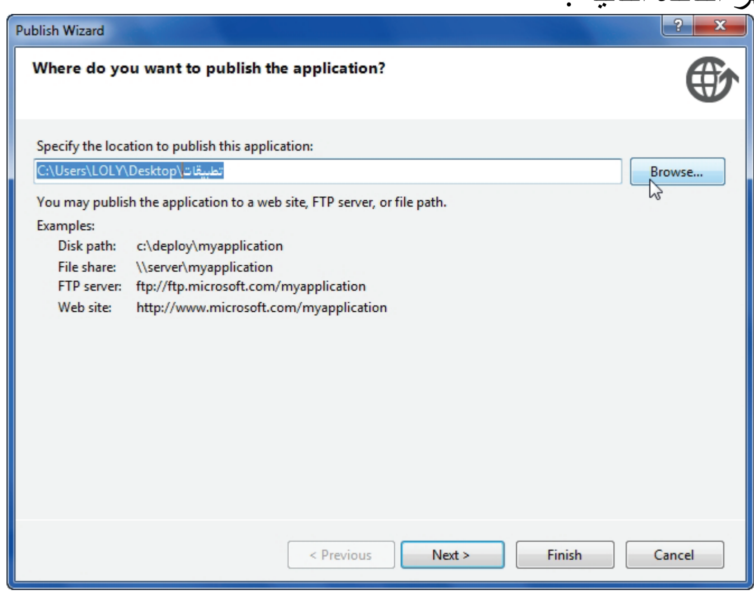


■ ثالثاً : إخراج البرنامج للعمل من خلال أيقونة Setup :

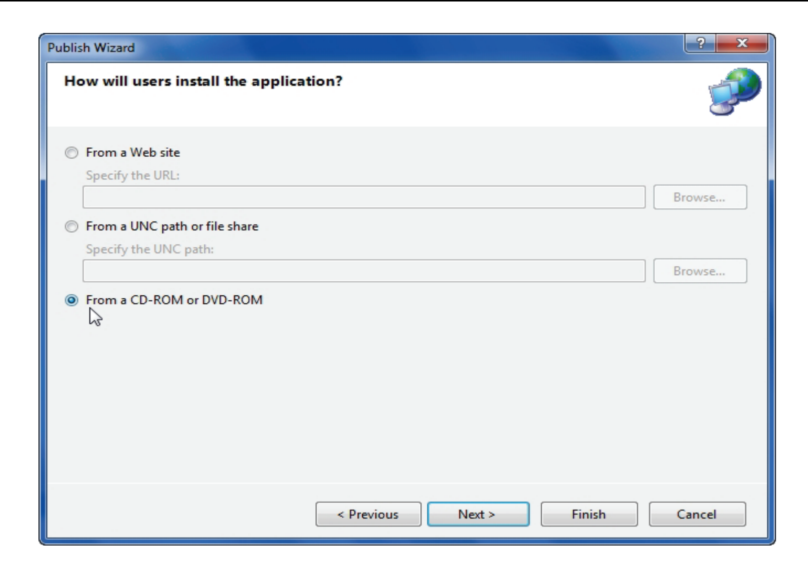
1 – من القائمة Build .. إختار الأمر Publish Web Browser ..



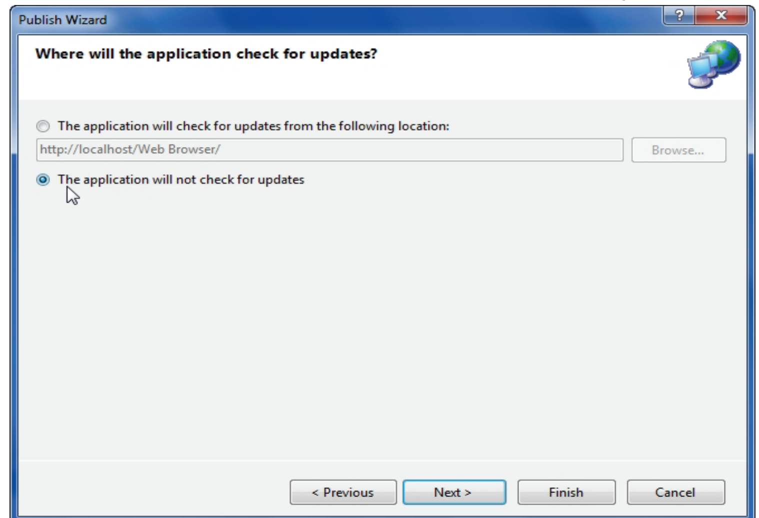
فتظهر النافذة التالية :



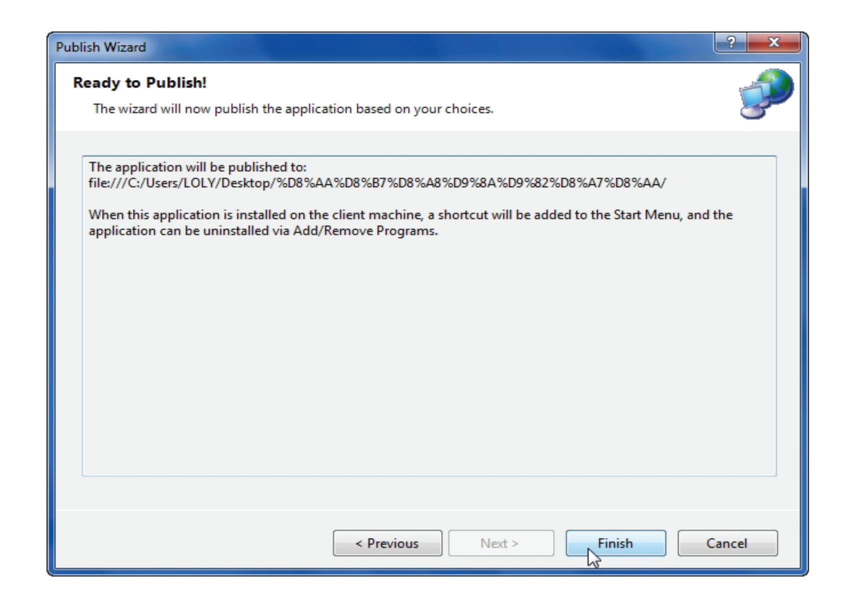
2- حدد مكان حفظ البرنامج بالضغط على المفتاح Browse .. ثم إضغط **Next** فتظهر النافذة التالية :



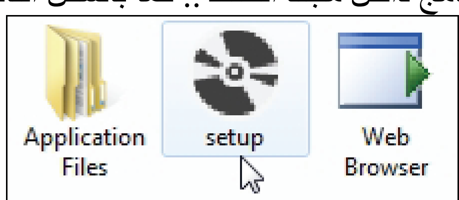
3 – نشط الإختيار From CD – Rom or DVD Rom .. ثم إضغط **Next** فتظهر النافذة التالية :



4 – نشط الإختيار The Application Will Not Check For Update ثم إضغط **Next** .. فتظهر النافذة التالية :



5– إضغط **Finish** .. للإنتهاء من إعداد البرنامج للنشر ..فتظهر الأيقونات الخاصة بالبرنامج داخل مجلد الحفظ .. كما بالشكل التالى :



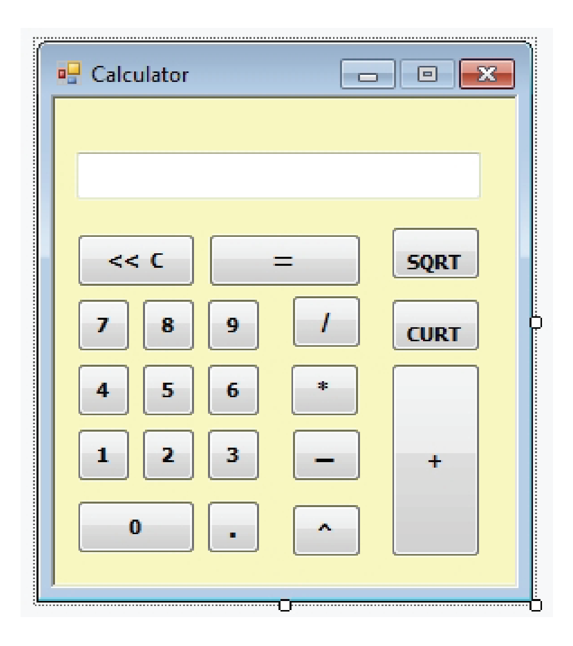
6 – قم بعمل Setup للبرنامج على جهاز الكمبيوتر .

■ تطبيق 2 : تصميم آله حاسبة Calculator

لتصميم برنامج آله حاسبة Calculator .. إتبع الخطوات التالية :

1 – إفتح مشروع جديد New Project .. ثم حدد اللون المناسب للخلفية .

2 – أضف Text Box ، ثم أضف عدد 20 مفتاح تحكم Button على النموذج Form .. وقم بتغيير الخاصية Text ، والخاصية Font لكل مفتاح .. وقم أيضاً بعمل التنسيقات اللازمة .. كما بالشكل التالى :



3 – إضغط على كل مفتاح مرتين .. وأكتب الأكواد التالية :

• **كود مفتاح رقم 0 :**

```
TextBox1.Text = TextBox1.Text + "0"
```

• **كود مفتاح رقم 1 :**

```
TextBox1.Text = TextBox1.Text + "1"
```

• **كود مفتاح رقم 2 :**

```
TextBox1.Text = TextBox1.Text + "2"
```

• **كود مفتاح رقم 3 :**

```
TextBox1.Text = TextBox1.Text + "3"
```

• **كود مفتاح رقم 4 :**

```
TextBox1.Text = TextBox1.Text + "4"
```

• **كود مفتاح رقم 5 :**

```
TextBox1.Text = TextBox1.Text + "5"
```

• **كود مفتاح رقم 6 :**

```
TextBox1.Text=TextBox1.Text+"6"
```

• **كود مفتاح رقم 7 :**

```
TextBox1.Text=TextBox1.Text+"7"
```

• **كود مفتاح رقم 8 :**

```
TextBox1.Text=TextBox1.Text+"8"
```

• **كود مفتاح رقم 9 :**

```
TextBox1.Text=TextBox1.Text+"9"
```

• **كود مفتاح رقم 10 :**

```
TextBox1.Text=TextBox1.Text+"10"
```

4 – قم بتعريف متغيرين للنموذج Form 1 .. الأول من النوع **Single** .. ليعبر عن الرقم الأول **S1** ، والرقم الثانى **S2** فى العملية الحسابية .. والمتغير الثانى من النوع **String** ليعبر عن ناتج العملية الحسابية **Op**.. وذلك كما بالكود التالى :

```
Dim s1, s2 As Single
Dim op AS String
```

5 – أكتب أكواد المفاتيح التالية :

• **كود مفتاح عملية الجمع** (+) :

```
S1 = Val(TextBox1.Text)
TextBox1.Clear()
OP = "+"
```

- **كود مفتاح عملية الطرح (-) :**

```
S1 = Val(TextBox1.Text)
TextBox1.Clear()
 OP = "-"
```

- **كود مفتاح عملية الضرب (*) :**

```
S1 = Val(TextBox1.Text)
TextBox1.Clear()
 OP = "*"
```

- **كود مفتاح عملية القسمة (/) :**

```
S1 = Val(TextBox1.Text)
TextBox1.Clear()
 OP = "/"
```

- **كود مفتاح العلامة العشرية (.) :**

```
TextBox1.Text = TextBox1.Text + "."
```

- **كود مفتاح الناتج يساوى (=) :**

```
S2 = Val(TextBox1.Text)
 TextBox1.Clear()
 If op = "+" Then
    TextBox1.Text = (S1 + S2)
 End If
 If Op = "-" Then
    TextBox1.Text = (S1 - S2)
 End If
 If Op = "*" Then
    TextBox1.Text = (S1 * S2)
 End If
 If op = "/" Then
    TextBox1.Text = (S1 / S2)
```

• كود مفتاح مسح العملية الحسابية (<< C) :

```
TextBox1.Clear()
TextBox1.Focus()
```

• كود مفتاح الجذر التربيعى (SQRT) :

```
Dim Sqrt As Double
Sqrt = TextBox1.Text ^ (1 / 2)
TextBox1.Text = Str(Sqrt)
```

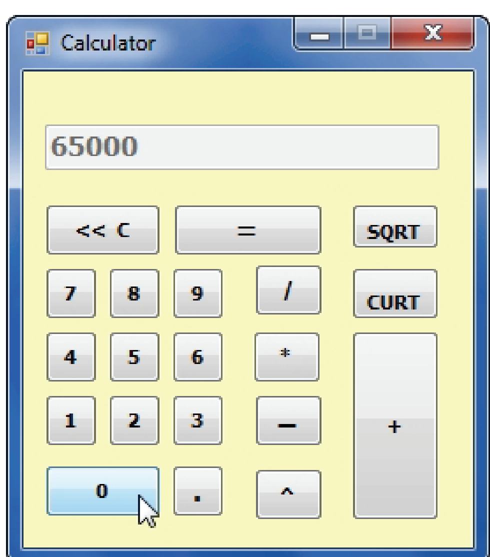
• كود مفتاح الجذر التكعيبى (CURT) :

```
Dim Curt As Double
Curt = TextBox1.Text ^ (1 / 3)
TextBox1.Text = Str(Curt)
```

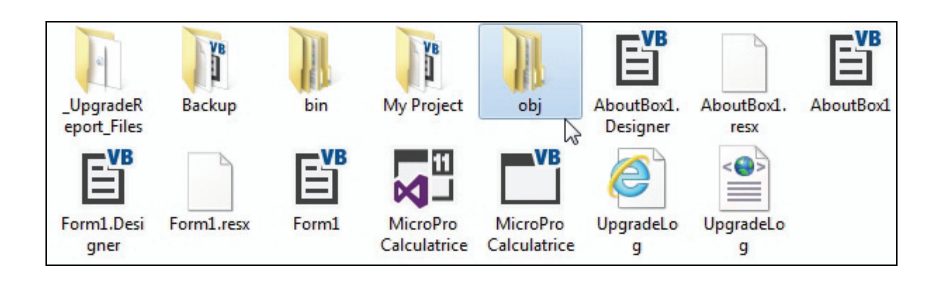


بالنسبة لكود الجذر التربيعى والتكعيبى .. تم تعريف متغير من النوع Double بأسم SQRT .. حيث أن الجذر التربيعى يساوى الرقم المطلوب إيجاد جذره التربيعى مضروب فى الرقم واحد أس (1/2) .. وكذلك الجذر التكعيبى يأخذ أسم CURT وهو الرقم المطلوب إيجاد جذره التكعيبى مضروب فى الرقم واحد أس (1/3) .

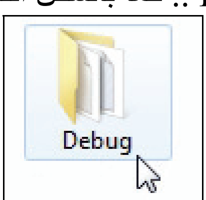
6 – شغل البرنامج .. بالضغط على مفتاح **F5** .. فيظهر البرنامج كما بالشكل التالى :



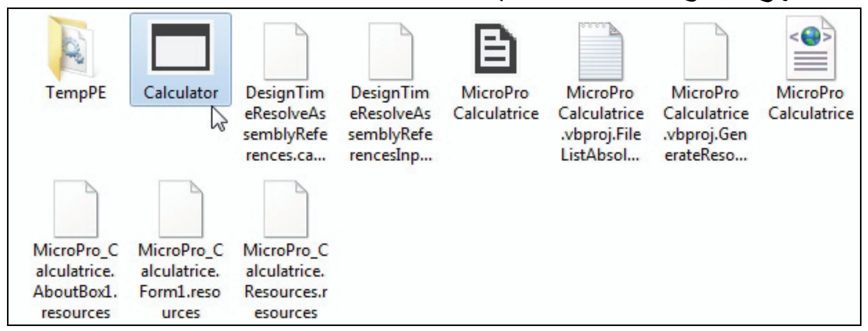
7 – قم بعمل أيقونة تشغيل بإمتداد الملف التنفيذى **EXE** .. وذلك بفتح ملف حفظ المشروع Calculator .. ثم إفتح المجلد المسمى Obj .. كما بالشكل التالى :



فيظهر المجلد المسمى Debug .. كما بالشكل التالى :



8 – تظهر مجموعة الملفات التالية :



9 – قم بنسخ الأيقونة **Calculator** .. ثم وضعها فى أى مكان تريده على جهاز الكمبيوتر .. ثم إضغط على الأيقونة **Calculator** مرتين ليتم فتحها وتشغيلها .. دون الحاجة إلى عمل Setup .



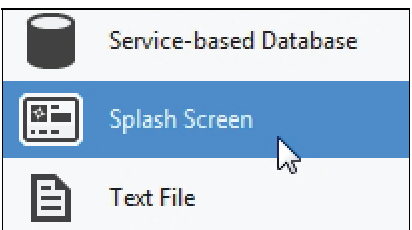
يمكن عمل أيقونات تنفيذية للبرامج بإمتداد EXE .. وذلك البرامج المكونة من Form واحد فقط .. وليس مجموعة Forms .. مثل برنامج الأله الحاسبة .

■ تطبيق 3 : تصميم نافذة إفتتاحية لواجهة البرامج Splash Screen

صمم برنامج يقوم بإظهار نافذة ترحيب عند بداية فتح البرنامج .. ولتنفيذ ذلك إتبع الخطوات التالية :

1 – إفتح مشروع جديد New Project .. بأسم Splash Scree .

2 – من القائمة Project .. إختار الأمر Add New Item .. فتظهر النافذة التالية :



إختار منها Splash Scree .. ثم إضغط Add .. فتظهر نافذة النموذج كما بالشكل التالى :



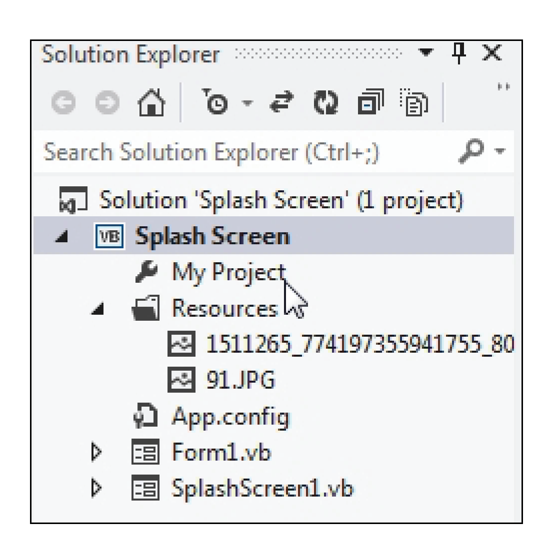


لا تقوم بحذف النص ، أو الصورة الموجودة بالنافذة السابقة بالمفتاح Delete لأن مــرتبط بها أكواد برمجية .. وأنما تغييرها من نافذة الخصائص Properties .

3 – قم بحذف النصوص الموجودة بالنافذة Splash Screen .. من الخاصية Text .

4 – غير الصورة الموجودة بالنافذة Splash Screen .. من الخاصية Background Image .

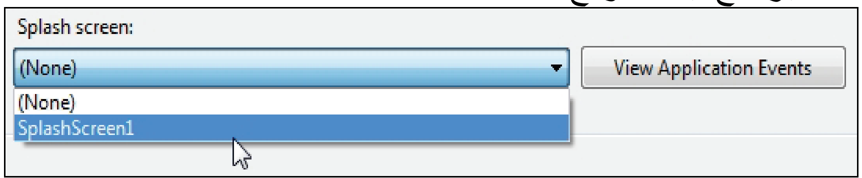
5 – إضغط مرتين على My Project .. من النافذة Solution Explorer الموجودة يمين نافذة البرنامج كما بالشكل التالى :



فتظهر النافذة التالية :

Configuration: N/A Platform: N/A
Assembly name: Splash Screen
Root namespace: Splash_Screen
Target framework: .NET Framework 4.5
Application type: Windows Forms Application
Startup form: Form1
Icon: (Default Icon)
Assembly Information... View Windows Settings
Enable application framework
Windows application framework properties
Enable XP visual styles
Make single instance application
Save My.Settings on Shutdown
Authentication mode: Windows
Shutdown mode: When startup form closes
Splash screen: SplashScreen1 View Application Events

6 – ومن النافذة السابقة إختار من القائمة المنسدلة Splash Screen .. الإختيار Splash Screen أيضاً .. لتظهر فى الواجهة الرئيسية للبرنامج قبل النموذج Form 1 ..



7 – شغل البرنامج .. بالضغط على مفتاح **F5** .. فتظهر نافذة الشاشة الإفتتاحية Splash Scerrn .. كما بالشكل التالى :



ثم بعد ذلك .. تظهر النافذة الرئيسية للبرنامج .. كما بالشكل التالى :





يمكنك الضغط بمفتاح الـ Mouse .. على نافذة الـ Splash Screen للدخول للنافذة الرئيسية للبرنامج ، أو الإنتظار قليلاً حتى تختفى هذه النافذة .

■ تطبيق 4 :تصميم نافذة لإظهار كلمة سر للبرنامج Password Login Form

لتصميم برنامج لإظهار كلمة سر تظهر قبل بداية البرنامج .. إتبع الخطوات التالية :

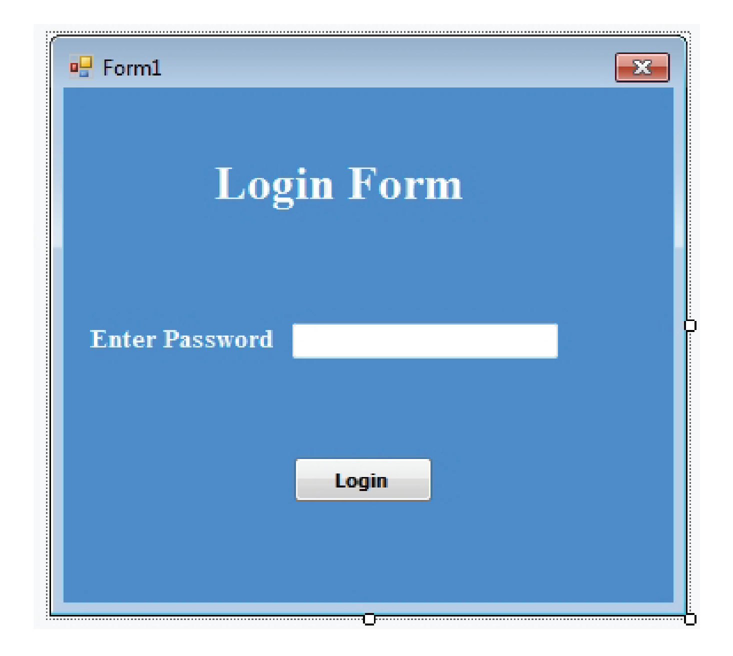
1 – إفتح مشروع جديد New Project .. بأسم Password Login .

2 – أضف Text Box ، ومفتاح Button ، عدد 2 Label لنافذة النموذج .. ثم إختار اللون المناسب لخلفية النموذج .

3 – غير الخاصية Text للعنوان الأول .. لتصبح Login Form ، والعنوان الثانى .. لتصبح Enter Password .

4 – غير الخاصية Text للمفتاح Button .. لتصبح Login .

5 – ضع العلامة الشاملة * للأداة Text Box .. فى الخاصية Password وذلك لتظهر كلمة السر على شكل نجوم عند إدخالها .. فتظهر نافذة النموذج كما بالشكل التالى :



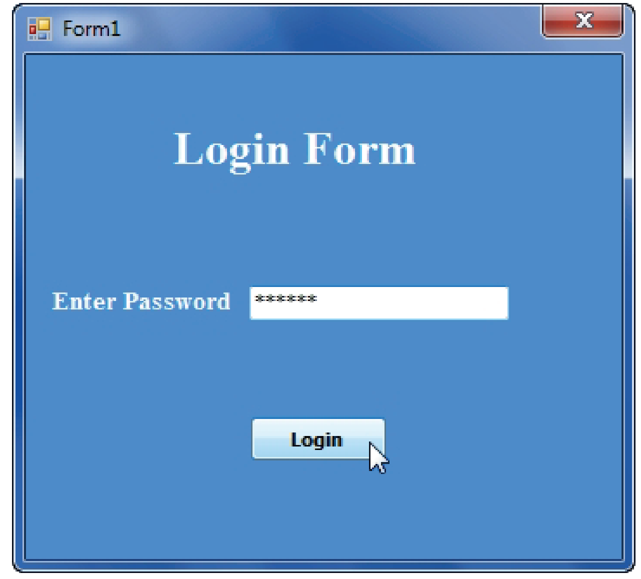
6 – إضغط مرتين على المفتاح Login .. وأكتب الكود التالى :

```
 If TextBox1.Text="visualbasic" Then
     Me.Hide()
     form2.show()
  Else
     MessageBox.Show("كلمة السر خاطئة من فضلك أعد المحاولة")
End If
```

7 – قم بفتح نموذج جديد Form 2 .. ونفذ بداخلة التطبيق السابق ، أو قم بإستدعاء التطبيق السابق ، ونسخ محتوياته .. كما بالشكل التالى :



8 – شغل البرنامج .. بالضغط على مفتاح **F5** .. فتظهر نافذة الـ Password كما بالشكل التالى :

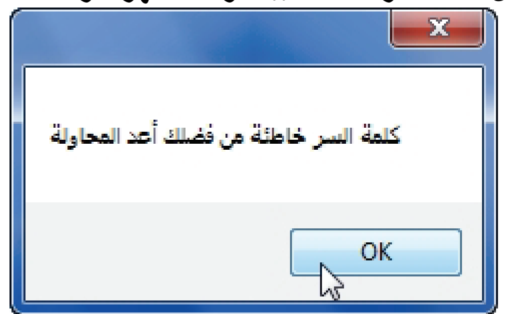


9 – إدخل كلمة السر وهى .. **visualbasic** .. ثم إضغط Login .. فتظهر النافذة الرئيسية للبرنامج .. كما بالشكل التالى :





فى حالة إدخال كلمة سر خاطئة .. سوف تظهر الرسالة التالية :

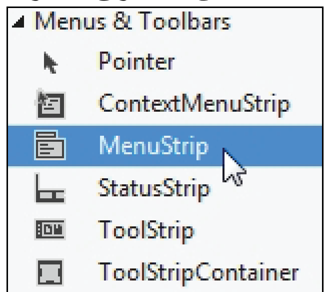


■ تطبيق 5 : تصميم شريط القوائم Menu Bar

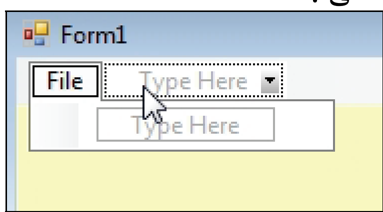
صمم نموذج لبرنامج يحتوى على شريط القوائم Menu Bar .. ولتنفيذ ذلك إتبع الخطوات التالية :

1 – إفتح مشروع جديد New Project .. بأسم Menu Bar .

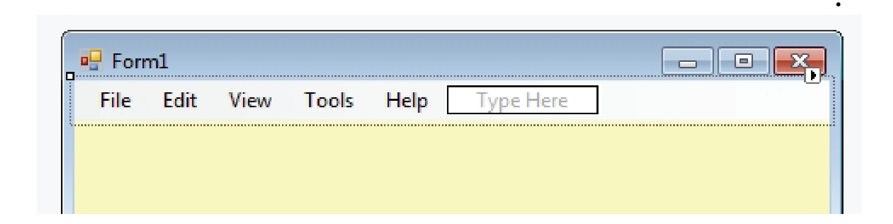
2 – أضف الأداة Menu Strip من صندوق الأدوات Tool Box ..



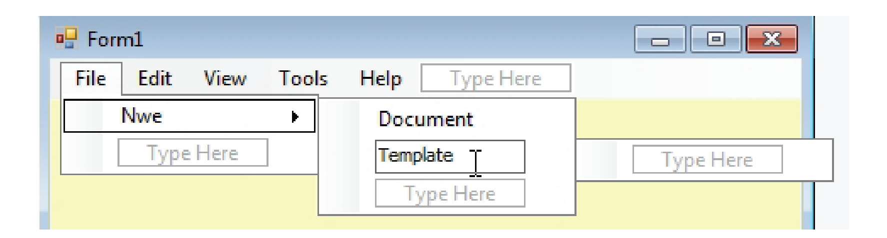
فتظهر كما بالشكل التالى :



3 – أكتب أسماء القوائم داخل المستطيل الفارغ لكل قائمة .. كما بالشكل التالى :

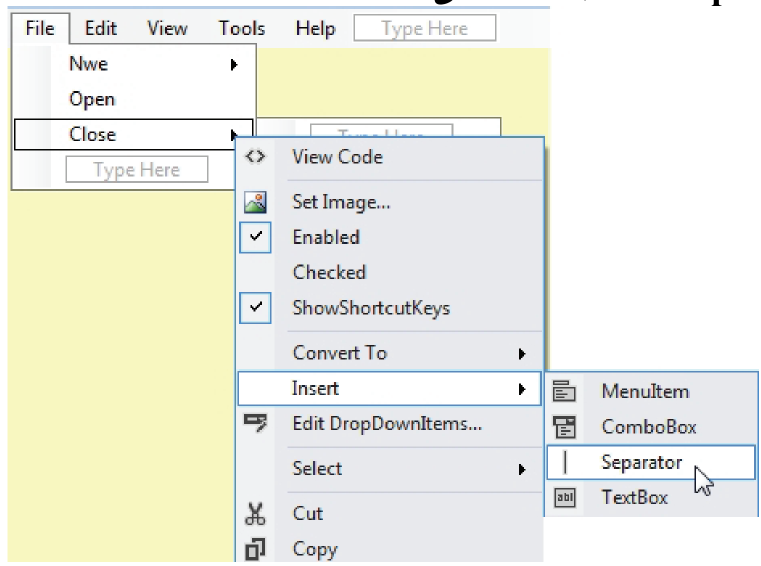


4 – إضغط على القائمة File .. ثم إدخل أسماء الأوامر داخل القائمة .. كما بالشكل التالى :

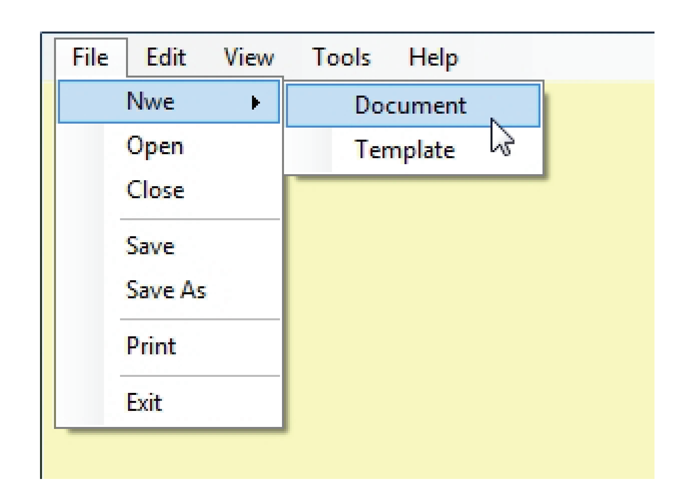




يمكن الفصل بين الأوامر المتشابهه ، والأوامر الأخرى داخل القائمة الواحدة بخط فاصل .. وذلك بالضغط بمفتاح الـ Mouse الأيمن ..فتظهر قائمة مختصرة .. إختار منها الأمر Insert .. ثم من القائمة الفرعية الأمر Separator .. كما بالشكل التالى :



فيظهر الشكل النهائى للقوائم .. كما بالشكل التالى :



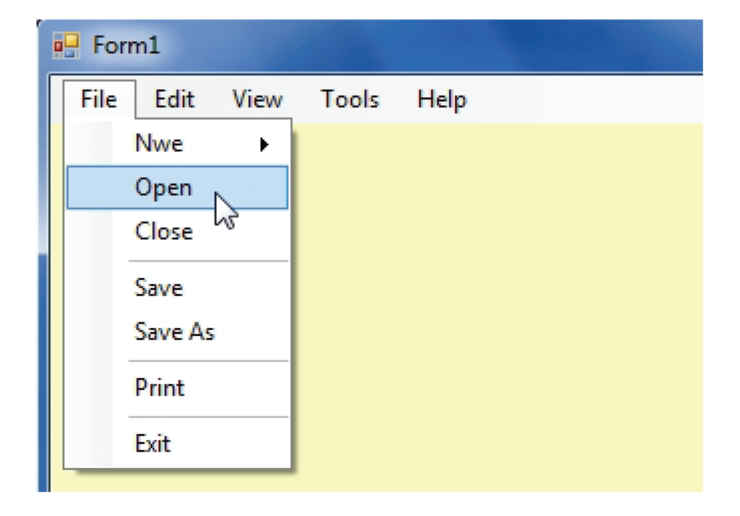
5 – أكتب الكود الخاص بكل أمر داخل كل قائمة .. وذلك بالضغط مرتين على أسم الأمر .. فمثلاً إضغط على الأمر **Open** .. وأكتب الكود التالى :

```
Dim DlgFile As New OpenFileDialog
DlgFile.ShowDialog()
```

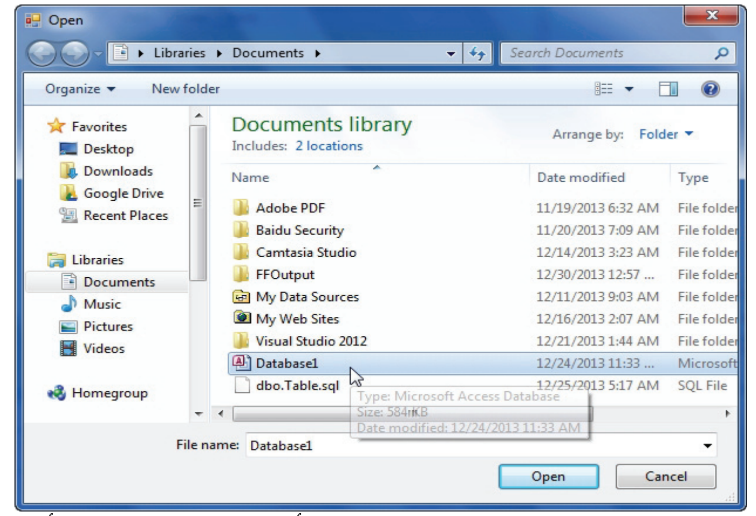
6 – أكتب الكود الخاص بالأمر **Exit** .. كما بالكود التالى :

```
Application.Exit
```

7 – شغل البرنامج .. بالضغط على مفتاح **F5** .. ثم إضغط على القائمة File ومنها الأمر Open ..



فتظهر النافذة التالية :



وبذلك يمكن تنفيذ العديد من القوائم ، والأوامر داخل القوائم لأى برنامج بنفس الطريقة .

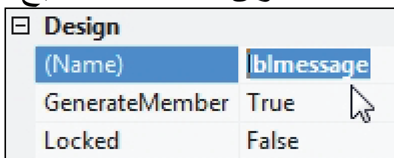
■ تطبيق 6 : تصميم نص متحرك داخل نافذة البرنامج

لتصميم شريط عنوان ، أو نص متحرك داخل نافذة البرنامج .. إتبع الخطوات التالية :

1 – إفتح مشروع جديد New Project .

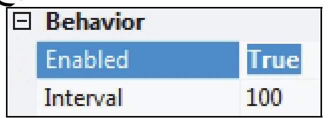
2 – أضف عنوان Label ، ثم أضف الأداة Timer .

3 – غير الخاصية Name للعنوان Label .. لتصبح lblmessage ..



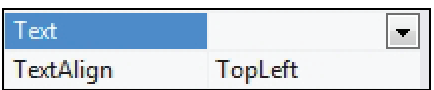
Design	
(Name)	lblmessage
GenerateMember	True
Locked	False

4 – غير الخاصية Enable للأداة Timer .. لتصبح True ..



Behavior	
Enabled	True
Interval	100

5 – إحذف النص Label 1 الموجود بالخاصية Text للأداة Label .. لتصبح فارغة



Text	
TextAlign	TopLeft

6 – إضغط مرتين على الأداة Label .. ثم أكتب الكود التالي :

```
Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As
  System.EventArgs) Handles Timer1.Tick
     Static Dim intCompteur As System.Int32 = 0
     Dim strMessage As System.String = "دار البراء لنشر وتوزيع الكتب والوسائط العلمية"
     If intCompteur = strMessage.Length Then
        lblmessage.Text = ""
        intCompteur = 0
     End If
     lblmessage.Text &= strMessage.Substring(intCompteur, 1)
     intComputer +=1
```

7 – غير حجم ، ولون الخط بالشكل المناسب للنص .

8 – شغل البرنامج .. بالضغط على مفتاح **F5** .. فيظهر النص المتحرك داخل نافذة البرنامج .. كما بالشكل التالى :



وفى النهاية

- كل ما عليك تطبيق الأمثلة والتطبيقات أكثر من مرة حتى تصل إلى مستوى الإحتراف المطلوب .. فمازال هناك العديد من الأفكار والأسرار للبرمجة بلغة الـ Visual Bsic .

- يمكنكم متابعة المزيد من الفيديوهات التعليمية الخاصة بإصدارات دار البراء ، والمؤلفات الخاصة بى .. من خلال الموقع التالى :

www.connect-eg.net

- كما يمكنكم متابعة أحدث الإصدارات والسلاسل التعليمة لدار البراء من خلال الموقع التالى :

www.daralbraa.com

- كما يمكنكم إرسال الأسئلة والإستفارات والإقتراحات .. من خلال صفحة الرسائل الخاصة بالموقع التالى :

www.connect-eg.net/Message

- كما يمكنكم المشاركة والتواصل من خلال الـ Facebook :

www.facebook.com/groups/717616484918161

المحتويات

المحتويات

الفصل الثالث

كتابة الكود داخل لغة

الفصل الرابع

التعامل مع المشروع Project

الفصل الخامس

التعامل مع صندوق الأدوات

الفصل السادس

المتغيرات والثوابت

الفصل السابع

الفصل الثامن

الجمل الشرطية والحلقات التكرارية

الفصل التاسع

الفصل العاشر

إكتشاف وتصحيح الأخطاء

الفصل الحادى عشر

قواعد البيانات

الفصل الثانى عشر

تطبيقات باستخدام لغة الـ Visual Basic



العنوان : 11 شارع د/محمد رأفت – محطة الرمل – الإسكندرية

تليفون وفاكس : 4838326 (03)(2+)

للاستعلام والمبيعات : 0101634294 (2+)

URL: www.daralbraa.com

Email: info@daralbraa.com